



Universidad Carlos III de Madrid

Ingeniería Técnica en Informática de Gestión

Proyecto Fin de Carrera

MD2: Sistema de búsqueda y composición  
de contenidos basados en IMS QTI

Autor: Iván Peláez López

Tutora: Carmen Luisa Padrón Nápoles

# **AGRADECIMIENTOS**

*No puedo dejar pasar esta oportunidad para agradecer su apoyo a las personas que han estado ahí conmigo durante mis estudios y a los que no han estado o nunca han creído que pudiera terminar mi carrera universitaria solo puedo dedicarles mi indiferencia.*

*En primer lugar quiero dar las gracias, como no, a mi madre, que ha estado en lo bueno y en lo malo, que me ha apoyado en todo momento y que ha aguantado mis malos momentos en época de exámenes. Mamá sabes que te quiero.*

*Después quiero agradecer a mis compañeros de clase los buenos momentos que me han hecho pasar y en especial quiero destacar a alguien que se ha convertido en un amigo y no solo en un compañero. Ese alguien es Antonio, mi eterno compañero de prácticas y con el que he pasado la mayor parte de las horas de mi carrera.*

*También quiero agradecer a mis amigos de siempre el haber estado ahí, para las risas y para los momentos duros, en especial a Rubén y a Himar, mis mejores amigos, con los que he compartido muchas horas de biblioteca y muchos momentos divertidos y felices de mi vida. Pero no me puedo olvidar del resto de mis amigos, mis gavilanes de siempre, de Álvaro, de Javi (hermanos os llevo en el corazón), de Mario, de Flores y otros tantos que pasaron por mi vida y de los que guardo buenos recuerdos.*

*Y por último quisiera agradecer a mi tutora, Carmen Padrón, la paciencia infinita que ha tenido conmigo y que ha permitido que pudiera terminar este proyecto. Probablemente no hubiera sido capaz de terminarlo con otro tutor y esto se lo agradeceré siempre.*

*A todos, gracias.*

<b>1. INTRODUCCION .....</b>	<b>8</b>
1.1 Objetivos .....	11
1.2 Planificación y método .....	13
1.3 Estructura del documento .....	15
<b>2. ESTADO DEL ARTE.....</b>	<b>16</b>
2.1 El concepto e-learning .....	16
2.2 Estandarización .....	18
2.3 Iniciativas de Estándares de e-learning .....	21
2.3.1 AICC, Aviation Industry CBT Committee .....	21
2.3.2 IEEE Learning Technologies Standards Committee (LTSC).....	22
2.3.3 IMS Global Learning Consortium, Inc. ....	23
2.3.4 ADL SCORM .....	27
2.4 Plataformas de enseñanza (LMS) .....	31
2.4.1 Moodle .....	32
2.4.2 .LRN .....	35
2.4.3 BlackBoard/WebCT .....	37
2.4.4. Otras plataformas LMS .....	40
<b>3. PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN .....</b>	<b>41</b>
3.1 Arquitectura de la solución.....	44
3.2. Tecnologías utilizadas .....	47
3.2.1 Struts .....	47
3.2.2. JDOM .....	49
3.2.3 XSLT .....	50
3.3 Especificaciones seguidas.....	51
3.3.1 IMS Question & Test Interoperability.....	51
3.3.2 IMS Content Packaging .....	59
3.3.3 IMS Learning Resources Meta-data .....	61
3.3.4 MD2 .....	63
<b>4. SOLUCIÓN DETALLADA .....</b>	<b>67</b>
4.1 Análisis.....	67
4.1.1 Análisis de Requisitos .....	67
4.1.2 Casos de uso .....	70
4.2 Diseño.....	85
4.2.1 Diseño de la capa de persistencia .....	86
4.2.2 Diseño de la capa de lógica de negocio .....	90
4.2.3 Diseño de la interfaz gráfica .....	96
<b>5. MANUAL DE USUARIO .....</b>	<b>98</b>
5.1. Acceso a la aplicación y menú principal .....	98
6.2. Administrar Usuarios.....	101
6.3. Administrar contenidos .....	109
6.3. Búsqueda de contenidos.....	117

5.4. Componer contenidos .....	122
5.5. Logout del sistema .....	123
5.6. Configuración de la aplicación.....	123
<b>6. GESTIÓN DEL PROYECTO.....</b>	<b>127</b>
6.1 Análisis de costes.....	131
6.1.1 Personal.....	131
6.1.2 Material e infraestructuras.....	132
6.1.3 Otros.....	133
6.1.4 Resumen de costes.....	134
<b>7. CONCLUSIONES .....</b>	<b>135</b>
7.1 Conclusiones del proyecto.....	135
7.2 Trabajo futuro y ampliaciones .....	137

---

## ÍNDICE DE FIGURAS

---

Figura 2.1. Principales participantes en el proceso de la enseñanza basada en <i>e-learning</i> [3] .....	18
Figura 2.2. Tipología de los estándares <i>e-learning</i> [5] .....	19
Figura 2.3. Colaboración entre los principales organismos de estandarización [3] .....	21
Figura 2.4. División en libros de la actual versión de SCORM [5] .....	28
Figura 2.5. Página de inicio de Moodle [22] .....	33
Figura 2.6. Página de administración de .LRN [20] .....	37
Figura 2.7. Página de inicio de la plataforma BlackBoard [26] .....	38
Figura 3.1. El patrón MVC. ....	44
Figura 3.2. Patrón MVC - Tecnologías Java .....	46
Figura 3.3. Implementación del patrón MVC con Struts [25] .....	47
Figura 3.4. Flujo de una operación con Struts .....	49
Figura 3.5. Ejemplo de pregunta de un examen tomado del sitio de la especificación IMS QTI [18] .....	52
Figura 3.6. Modelo de objetos de IMS QTI [18] .....	53
Figura 3.7. Relación entre los principales objetos de datos de IMS QTI 1.2 [18] .....	54
Figura 3.8. Raíz del documento XML de la especificación IMS QTI [18] .....	55
Figura 3.9. Esquema XML del objeto item [18] .....	56
Figura 3.10. Esquema XML del objeto <i>presentation</i> [18] .....	56
Figura 3.11. Esquema XML del objeto <i>response_lid</i> [18] .....	58
Figura 3.12. Ejemplo de pregunta con visualización <i>render_slider</i> [18] .....	59
Figura 3.13. Esquema de un manifiesto [15] .....	60
Figura 3.14. Ejemplo de manifiesto XML [15] .....	61
Figura 3.15. Vista del Modelo MD2 .....	65
Figura 4.1. Diagrama caso de uso Login .....	70
Figura 4.2. Diagrama de actividad del caso de uso Login .....	71
Figura 4.3. Diagrama del caso de uso Administrar repositorio .....	71
Figura 4.4. Diagrama de actividad de Alta contenido .....	72
Figura 4.5. Diagrama de actividad del caso de uso Baja contenido .....	72
Figura 4.6. Diagrama de actividad del caso de uso Crear Descriptor .....	73
Figura 4.7. Diagrama de actividad del caso de uso Añadir Descriptor .....	73
Figura 4.8. Diagrama de actividad del caso de uso Listado de contenidos .....	74
Figura 4.9. Diagrama del caso de uso Administrar usuarios .....	76
Figura 4.10. Diagrama de actividad del caso de uso Alta usuario .....	76
Figura 4.11. Diagrama de actividad del caso de uso Baja Usuario .....	77
Figura 4.12. Diagrama de actividad del caso de uso Modificar Usuario .....	77
Figura 4.13. Diagrama de actividad del caso de uso Listar Usuarios .....	78
Figura 4.14. Diagrama de caso de uso de Realizar Búsquedas .....	79

Figura 4.15. Diagrama de actividad del caso de uso Localizar Materiales.....	80
Figura 4.16. Diagrama de actividad del caso de uso Visualizar material .....	80
Figura 4.17. Diagrama de actividad del caso de uso Descargar material.....	81
Figura 4.18. Diagrama de actividad del caso de uso Editar material .....	81
Figura 4.19. Diagrama del caso de uso Realizar composiciones .....	83
Figura 4.20. Diagrama de actividad del caso de uso Componer un material ...	84
Figura 4.21. Diagrama de actividad del caso de uso Logout .....	84
Figura 4.22. Esquema Entidad/Relación de la base de datos.....	87
Figura 4.23. Esquema del modelo Relacional de la base de datos.....	89
Figura 4.24. Modelo de clases del sistema.....	91
Figura 4.25. Diagrama de secuencia del sistema.....	93
Figura 4.26. Diagrama de colaboración de los casos de uso Login y Logout.....	94
Figura 4.27. Diagrama de colaboración del caso de uso Administrar Usuarios	94
Figura 4.28. Diagrama de colaboración del caso de uso Realizar Búsquedas...	95
Figura 4.29. Diagrama de colaboración del caso de uso Realizar composiciones .....	95
Figura 4.30. Diagrama de colaboración del caso de uso Administrar Repositorio .....	96
Figura 4.31. Interfaz de ejemplo donde se observan las partes explicadas.....	97
Figura 5.1. Formulario de login de la aplicación .....	99
Figura 5.2. Página principal del usuario Administrador.....	100
Figura 5.3. Página para la administración de usuarios .....	101
Figura 5.4. Formulario de nuevo usuario .....	102
Figura 5.5. Página de usuario insertado correctamente .....	103
Figura 5.6. Pantalla de error en una operación.....	103
Figura 5.7. Página para buscar usuario a eliminar .....	104
Figura 5.8. Página para confirmar el borrado de un usuario.....	105
Figura 5.9. Página de usuario borrado correctamente .....	105
Figura 5.10. Página para buscar el usuario a modificar .....	106
Figura 5.11. Página para modificar los datos del usuario .....	107
Figura 5.12. Página de usuario modificado correctamente .....	108
Figura 5.13. Página que muestra el listado de usuarios.....	109
Figura 5.14. Página para administrar los contenidos.....	110
Figura 5.15. Buscador de contenidos a borrar .....	110
Figura 5.16. Listado de contenidos que se pueden borrar.....	111
Figura 5.17. Página para confirmar el borrado de los contenidos.....	112
Figura 5.18. Formulario par subir un contenido .....	113
Figura 5.19. Menú para añadir un descriptor al contenido.....	114
Figura 5.20. Formulario para subir un fichero de descriptores.....	115
Figura 5.21. Formulario para crear descriptores del contenido .....	116
Figura 5.22. Formulario para crear descriptores con un combo desplegado..	116
Figura 5.23. Página de confirmación para la subida de ficheros .....	117
Figura 5.24. Buscador de contenidos educativos.....	118
Figura 5.25. Listado de contenidos buscados .....	119
Figura 5.26. Ejemplo de un contenido visualizado.....	120
Figura 5.27. Ejemplo de descarga de contenidos.....	121
Figura 5.28. Página donde se muestra el nuevo contenido compuesto.....	122
Figura 6.1. Planificación de tareas del proyecto.....	128
Figura 6.2. Diagrama Gantt del proyecto.....	130
Figura 6.3. Costes del personal implicado en el proyecto.....	132

Figura 6.4. Resumen de días y costes totales del personal ..... 132

Figura 6.5. Costes de materiales e infraestructuras ..... 133

Figura 6.6. Otros gastos..... 133

Figura 6.7. Resumen de costes del proyecto..... 134

---

## 1. INTRODUCCION

---

Los contenidos educativos en formato digital, son una parte fundamental dentro de los nuevos modelos de enseñanza basados en *e-learning*. El concepto de *e-learning* se define de muchas formas diferentes fundamentalmente debido a que los actores que de él hacen uso son muy diversos, cada uno con su idiosincrasia y su ámbito de aplicación.

Desde la perspectiva de su concepción y desarrollo como herramienta formativa, los sistemas de *e-learning* tienen una dualidad pedagógica y tecnológica. Pedagógica en cuanto a que estos sistemas no deben ser meros contenedores de información digital, sino que ésta debe ser transmitida de acuerdo a unos modelos y patrones pedagógicamente definidos para afrontar los retos de estos nuevos contextos. Tecnológica en cuanto que todo el proceso de enseñanza-aprendizaje se sustenta en aplicaciones software, principalmente desarrolladas en ambientes web, lo que le vale a estos sistemas el sobrenombre de plataformas de formación.

Desde la perspectiva de su uso se podría distinguir la visión que tienen sus usuarios finales, que con independencia de su madurez y formación, verán al sistema *e-learning* como una fuente de servicios para alcanzar su cometido formativo. No obstante, también es factible diferenciar una visión de organización, en la que se definen el alcance y los objetivos buscados con la formación basada en estos sistemas, distinguiéndose una visión académica y una visión empresarial.



Otros autores acotan más el alcance del *e-learning* reduciéndolo exclusivamente al ámbito de Internet, definido como: "*el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento*" [7]. Está basado en tres criterios fundamentales:

1. El *e-learning* trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.
2. Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.
3. Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación".

En las definiciones que se puedan dar, se acaba haciendo mención explícita o implícita a lo que se viene llamando el triángulo del *e-learning*, formado por la **tecnología** (plataformas, campus virtuales...), los **contenidos** (calidad y estructuración de los mismos se toman como elementos capitales para el éxito de una iniciativa de e-formación o aprendizaje basado en e-learning) y los **servicios** (siendo el elemento más variopinto que engloba la acción de los profesores, elementos de gestión, elementos de comunicación, elementos de evaluación...). Variando el peso de estos tres componentes se obtienen diferentes modelos de e-formación, de igual forma que variando las variables y recursos con los que cuenta un profesor se obtienen diferentes políticas de docencia presencial.

En la práctica, para llevar a cabo un programa de formación o de aprendizaje basado en *e-learning*, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea o LMS (*Learning Management Systems*) y, por otro lado, las que se utilizan para la gestión de los contenidos digitales o LCMS (*Learning Content Management Systems*).

Para poder hacer realidad esta nueva forma de crear contenidos, y debido a la heterogeneidad de plataformas educativas y de los sistemas de enseñanza en línea (es decir de los LMS), es necesaria la existencia de recomendaciones y estándares ampliamente aceptados que posibiliten la reutilización de los contenidos y su interoperabilidad entre diferentes sistemas.

## Estándares

En *e-learning*, una de las principales funciones de los estándares es servir como facilitadores de la durabilidad y de la reutilización en el tiempo de los contenidos y de la interoperabilidad, es decir, facilitar el intercambio de los contenidos entre diversas plataformas y sistemas. Hay que evitar caer en el error de ver el estándar como un limitador de la iniciativa o creatividad personal. En muchos casos, cuando los educadores oyen la palabra estándar suelen tener una reacción adversa, ya que tienden a considerar que es una norma de obligado cumplimiento que coartará su creatividad respecto a la creación de nuevos cursos, o su forma habitual de planificar una acción formativa o una clase. Otra circunstancia es considerar que su uso es sólo en educación a distancia y que no son útiles para otros planteamientos educativos. Esto no es cierto, ya que la existencia de contenidos educativos reutilizables puede ser de gran ayuda para simplificar el trabajo de los docentes, aunque lo utilicen en educación presencial o en un formato mixto presencial-web (llamado *blended learning* o *b-learning*).

Existen multitud de ventajas asociadas a la utilización generalizada de estándares de *e-learning* para todas las partes implicadas en el proceso de aprendizaje. Entre ellas cabe mencionar las siguientes [8]:

- Desde el punto de vista del de los clientes o consumidores tanto institucionales como individuales, los estándares evitan quedarse atrapado por las tecnologías propietarias. Los costes se reducen al sustituir los desarrollos propios por tecnología “plug and play” de modo que, por ejemplo, una institución pueda cambiar de LMS sin tener que empezar desde el principio perdiendo toda o gran parte de la información que ya tenía en su LMS anterior.
- Desde el punto de vista de los vendedores de aplicaciones, la existencia de métodos estandarizados de comunicación entre sistemas simplifica la integración de diferentes productos. Esto redundará en una reducción de los costes de desarrollo e incrementa el mercado potencial para las aplicaciones.
- Desde el punto de vista de los productores de contenidos educativos, los estándares permiten que el formato de producción sea único y pueda ser utilizado en cualquier plataforma de distribución. Más aún, un mercado más amplio para los contenidos educativos permite a los creadores realizar inversiones en producción de contenidos, aumentando la oferta y la calidad de éstos, incluso en áreas altamente especializadas. Además la existencia de estándares facilita su labor, al tener acceso a almacenes de contenidos reutilizables, y les permite la creación de contenidos modulares de más fácil mantenimiento y actualización.

- Desde el punto de vista de los alumnos, los estándares implican mayor posibilidad de elección del producto educativo. Además implican que los resultados de su aprendizaje (créditos o certificados) tengan mayor portabilidad.

En otros trabajos se destacan las ventajas y propiedades beneficiosas que se obtienen con la aplicación de los estándares [2].

- **Interoperabilidad.** Que se pueda intercambiar y mezclar contenido de múltiples fuentes y se pueda usar directamente en distintos sistemas. Que sistemas diferentes puedan comunicarse, intercambiar información e interactuar de forma transparente.
- **Reusabilidad.** Que el contenido pueda ser agrupado, desagrupado reutilizado de forma rápida y sencilla. Que los objetos de contenido puedan ensamblarse y utilizarse en un contexto distinto a aquél para el que fueron inicialmente diseñados.
- **Gestionabilidad.** Que el sistema pueda obtener y trazar la información adecuada sobre el usuario y el contenido.
- **Accesibilidad.** Que un usuario pueda acceder el contenido apropiado en el momento justo y en el dispositivo correcto.
- **Durabilidad.** Que los consumidores no queden atrapados en una tecnología propietaria de una determinada empresa. Que no haya que hacer una inversión significativa para lograr la reutilización o la interoperabilidad.
- **Escalabilidad.** Que las tecnologías puedan configurarse para aumentar la funcionalidad de modo que se pueda dar servicio a más usuarios respondiendo a las necesidades de la institución, y que esto no exija un esfuerzo económico desproporcionado.

## 1.1 Objetivos

El objetivo principal del proyecto es realizar una aplicación web que permita la búsqueda, visualización y composición de contenidos educativos, que cumplan un determinado estándar.

Para ello se realizará un estudio previo sobre los estándares actuales de *e-learning* atendiendo a los distintos elementos que infieren en ella como tecnología, plataformas y pedagogía, entre otros.

Como principal función de la aplicación será el hecho de permitir componer materiales a partir de contenidos previamente almacenados o que puedan ser subidos a la aplicación.

Para permitir la composición, es necesario localizar los componentes más adecuados para una situación instructiva en particular. La búsqueda se realizará en base a unos descriptores de los contenidos que incluyen objetivos de aprendizaje definidos a partir de los conceptos relacionados con la temática a tratar en el contenido. La información sobre temáticas, objetivos y conceptos está definida de acuerdo con los planes de estudios de ACM (Association for Computing Machinery) para la titulación Computer Science (CS) o Informática. Los descriptores de contenidos seguirán la especificación IMS Learning Resource Meta-data Specification versión 1.2.

La visualización de los contenidos recuperados se realizará bajo la especificación de visualización y entrega IMS QTI (Question and Test Interoperability) versión 1.2, y la composición de un material estará determinada según un fichero `imsmanifest.xml` acorde con dicha especificación y el estándar IMS Content Package. Ambas son propuestas que disponen de información suficiente para crear herramientas para la manipulación de contenidos con relativa facilidad.

Así mismo, la aplicación permitirá tanto la gestión de los usuarios de la aplicación, como la gestión de los contenidos.

Por tanto el proyecto se enmarca dentro del área de *e-learning*, orientado al desarrollo de materiales didácticos por parte de especialistas en esta área. La creación de dichos materiales, en esta aplicación se producirá a través de la composición de contenidos.

Por último destacar un segundo objetivo del proyecto, que es la profundización en el desarrollo de aplicaciones web, y concretamente en el desarrollo de aplicaciones J2EE debido al auge de esta tecnología en el mundo empresarial.

## ***1.2 Planificación y método***

La metodología seguida en el desarrollo del proyecto es el método en cascada, que es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior, habiendo realizado las siguientes fases:

1. Recogida de requisitos de usuario. En esta fase, mediante reuniones con el personal conocedor de las necesidades a cubrir, se extraen los requisitos de los usuarios que debe satisfacer el sistema. En este caso las reuniones se mantuvieron con la tutora del proyecto.
2. Análisis de requisitos de usuario. Tras la recogida de los requisitos es preciso realizar una ordenación y análisis de los mismos, con el objetivo de facilitar la extracción de conclusiones en posteriores fases.
3. Diseño de alto nivel. En esta fase se definen los componentes existentes en el sistema de forma abstracta pero suficientemente detallados para posteriormente mantener una trazabilidad con los artefactos desarrollados.
4. Diseño detallado. Para cada componente general es preciso realizar una descripción precisa de los elementos que contendrá, definiendo su comportamiento, la información entrante y el resultado a generar. Esta información sirve para independizar la fase implementación del equipo de diseño, de forma que se puedan separar responsabilidades garantizando la corrección del producto final.
5. Implementación. En esta fase se genera el código fuente y los elementos necesarios par el correcto funcionamiento del sistema. Es habitual aislar esta fase del resto, optando incluso por la subcontratación de recursos. El objetivo de esta separación es evitar influencias negativas de una de las fases más complejas del proceso, aunque la existencia de frameworks garantizan que este proceso sea eficaz.
6. Pruebas. Una vez obtenido el producto desarrollado es necesario realizar pruebas que garanticen la calidad del mismo. La definición de dichas pruebas debe realizarse en base a los requisitos del sistema, no a la codificación realizada. Preferiblemente estas pruebas serán ejecutadas por personal independiente a la fase de implementación. Como experiencia, la aplicación ha sido utilizada en la asignatura "Enseñanza asistida por ordenador" del quinto curso de la titulación Ingeniería Informática.
7. Implantación. Tras haber comprobado la factibilidad del uso y sobretudo la utilidad del sistema, y el cumplimiento de los requisitos de usuario se procede a la implantación del sistema en el lugar donde

llevará a cabo su labor. No es conveniente subestimar esta fase pues puede provocar la invalidez del proyecto tras haber invertido en él gran cantidad de recursos.

No obstante, debido a ciertas particularidades del proyecto - fundamentalmente el carácter académico y las propiedades del equipo de desarrollo, formado por una única persona-, se han producido alteraciones al modelo de trabajo habitual:

- El diseño de alto nivel se mantuvo excesivamente abstracto, fundamentalmente por el desconocimiento, a nivel técnico, de las tecnologías que se iban a emplear. A pesar de que las tecnologías concretas no deberían influir en el diseño de alto nivel, es cierto que un conocimiento mayor de las alternativas existentes hubiera permitido profundizar más en ciertos aspectos sin temor a no encontrar respaldo en las posteriores fases.
- El diseño detallado ha sido elaborado paralelamente a la implementación, lo cual es habitual hasta cierto grado en muchos proyectos; aunque por el carácter académico de esta aplicación no ha habido otra opción que simultanear las dos fases.
- Las pruebas han sido definidas por el propio equipo de desarrollo, lo cual resulta inadecuado y no garantiza la eficacia de las mismas. La participación de la tutora del proyecto en las pruebas ha posibilitado otro enfoque distinto al desarrollador y por tanto garantizar un mínimo de objetividad
- En la definición del proyecto se establece que el entorno será el académico en el cual se realizará la implantación. En este caso el cliente, es a la vez el tutor del proyecto y por tanto la persona que establece restricciones al respecto. La implantación ha sido llevada a cabo por el propio equipo de desarrollo, garantizando el correcto funcionamiento en dicho entorno.

Una de las herramientas que facilitan la consecución del éxito en el desarrollo de proyectos informáticos es la planificación de tareas. La elaboración de dicha planificación puede llevarse a cabo en distintas fases del proyecto, en función de aspectos como la experiencia del personal responsable de realizar la planificación o el conocimiento sobre la temática asociada al proyecto. Es habitual, y símbolo de eficacia, el cambio en la planificación actual desarrollada; es preciso alterar la planificación en función de los distintos aspectos que surgen a lo largo del proyecto, demostrando dinamismo y capacidad de adaptación.

La planificación completa y la gestión del proyecto se describirán con más detalle en el capítulo 4.

### ***1.3 Estructura del documento***

El contenido del presente documento ha sido estructurado de la siguiente forma:

1. **Introducción:** en este capítulo se realiza una descripción de la situación que ha impulsado el desarrollo de este proyecto, se establecen cuáles son los principales objetivos y se presenta un breve resumen del trabajo realizado.
2. **Estado del arte:** este capítulo se analiza el panorama actual de los sistemas dedicados al *e-learning*, presentando proyectos relacionados, así como las especificaciones que intentan estandarizar la cuestión.
3. **Planteamiento del problema y solución:** en este capítulo se describen cuáles son los problemas que el proyecto pretende subsanar, y se detallan las tecnologías utilizadas para ello.
4. **Solución detallada:** en este capítulo se describe la solución propuesta a un nivel más profundo, tanto en el análisis realizado como en el diseño de la solución.
5. **Manual de usuario:** en este capítulo se muestra un pequeño manual de uso de la aplicación para mostrar al usuario su funcionamiento.
6. **Gestión del proyecto:** en este capítulo se detalla la planificación del proyecto, describiendo las distintas tareas a realizar y el tiempo planificado para cada una de ellas.
7. **Conclusiones:** por último este capítulo presenta reflexiones sobre la consecución de los objetivos marcados y esboza posibles líneas de trabajo con las que continuar a partir de este proyecto.

---

## 2. ESTADO DEL ARTE

---

### *2.1 El concepto e-learning*

*E-learning* es un tipo de enseñanza que mejora el conocimiento basándose en el uso de nuevas tecnologías y especialmente en el mundo de Internet. Desde la irrupción de Internet en nuestras vidas, las posibilidades de acceso a la formación se han ido incrementando en la medida en que la Red nos va permitiendo acceder a más personas y ofrecer ambientes de aprendizaje más complejos y elaborados.

No obstante, el término *e-learning* es algo relativamente nuevo en el mundo de la informática y la educación y es posible encontrar varias definiciones distintas. Básicamente, podría decirse que *e-learning* es todo aprendizaje, entrenamiento o educación que se facilita mediante el uso de las tecnologías telemáticas, especialmente las redes basadas en la tecnología de Internet. El *e-learning* es un método de educación a distancia, pero no toda educación a distancia es *e-learning*.

En los inicios del desarrollo de este campo, se pudo plantear la llegada del *e-learning* como un sustituto de la docencia presencial que hasta entonces había existido. El *e-learning* en su evolución debería acabar por extinguir este otro tipo de educación tradicional. Lo cierto es que esto no ha ocurrido mayoritariamente, en su lugar ha surgido otro modo de entender la llegada de las nuevas tecnologías a la educación, el *Blended Learning (b-learning)*. El *b-learning* es el enriquecimiento de la formación presencial mediante el uso de



las capacidades de *e-learning*. La formación presencial puede ser combinada con un sistema de aprendizaje en línea que sirve a los alumnos para acceder en cualquier momento y en cualquier lugar a multitud de contenidos multimedia, herramientas de comunicación, mensajería instantánea, etcétera.

Una de las principales ventajas del *e-learning* es la facilidad de acceso, La formación puede llegar a más personas, puesto que desaparecen las barreras espacio-temporales. De esta forma, personas que antes tenían dificultades para estar en contacto continuo con los procesos de formación, por problemas de desplazamiento al centro donde se imparten los cursos, por escasez de tiempo, por incapacidad física para asistir a clase, por vivir en pequeñas aldeas poco comunicadas con el exterior, etcétera, tienen ahora todo un abanico de posibilidades a su disposición para una formación continua. No obstante, *e-learning* mantiene más ventajas con respecto a las técnicas de aprendizaje clásicas entre las que destacan:

- Elimina las distancias físicas: se utilizan herramientas como correo electrónico, foro o chat par establecer la comunicación entre los participantes.
- Permite flexibilidad horaria: el alumno accede en el momento que dispone de tiempo.
- Aumenta el número de destinatarios: esta modalidad de formación se puede dirigir a una audiencia mucho más amplia.
- Aumento de escalabilidad de los destinatarios: las soluciones de *e-learning* son altamente escalables. Los programas puedes pasar de 10 participantes a 100 o incluso 10.000 con un bajo coste.
- Favorece la interacción: los alumnos pueden comunicarse unos con otros, con el tutor y con los recursos on-line disponibles en Internet.
- Disposición de recursos *on-line* y multimedia: Internet proporciona acceso instantáneo e ilimitado a una gran cantidad de recursos multimedia, como textos, gráficos, audio, vídeo, animaciones, etcétera.
- El contenido es más oportuno: por estar disponible en web, el *e-learning* pude actualizarse instantáneamente. Esto proporciona la capacidad de perfeccionar el contenido más fácil y rápidamente y distribuir de forma inmediata la nueva información.
- Aporta mayor facilidad para la interacción multicultural, ya que, alumnos procedentes de distintas culturas pueden coincidir en un mismo curso.
- Aprovecha la gran capacidad de procesamiento de las computadoras para ofrecer al alumno una enseñanza más personalizada.

El *e-learning* está soportado por un software que implementa sus técnicas de docencia, lo dota de las herramientas adecuadas y distribuye los contenidos. Entre los productos que apoyan el *e-learning* están los LMS.

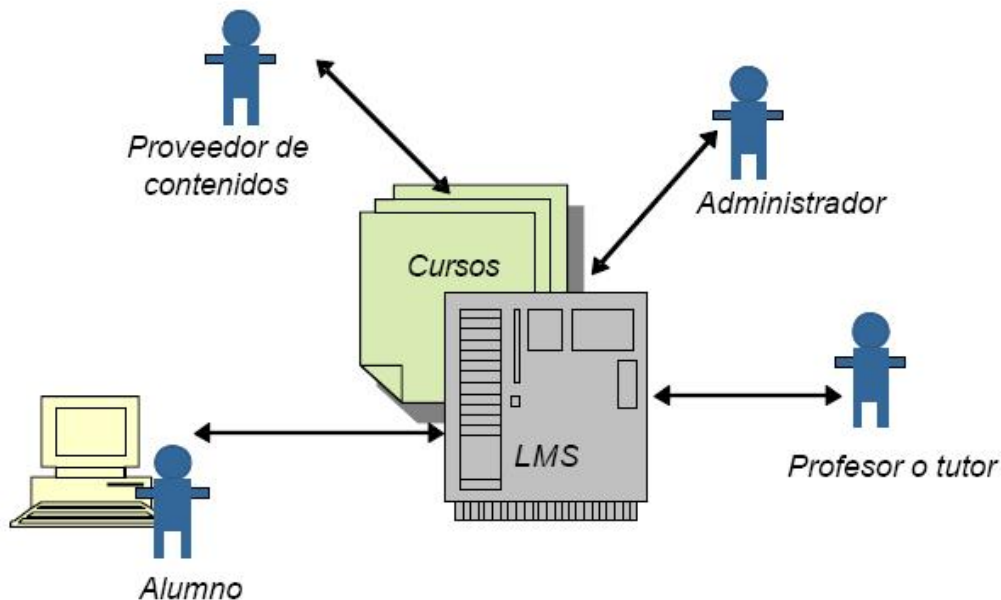


Figura 2.1. Principales participantes en el proceso de la enseñanza basada en *e-learning* [3]

LMS (Learning Management System) es el software que automatiza la administración de acciones de formación o contenidos. Un LMS registra usuarios, organiza los diferentes cursos, almacena datos sobre los usuarios o también provee informes para la gestión. Además suministra al profesor un mecanismo para distribuir contenido, supervisar la participación de los estudiantes y evaluar su desempeño. También suelen ofrecer a los estudiantes el uso de mecanismos de interacción como foros de discusión o chats. Un LMS, generalmente, incluye posibilidades de autoría (crear sus propios contenidos), pero se centra en gestionar contenidos creados por una gran variedad de fuentes diferentes.

A continuación se analiza la estandarización y los estándares disponibles en el campo del *e-learning*.

## 2.2 Estandarización

En el mercado existen multitud de LMS de muchos fabricantes distintos. Por ello se hace necesaria una normativa que compatibilice los distintos sistemas y cursos a fin de lograr dos objetivos:

- Que un curso de cualquier fabricante pueda ser cargado en cualquier LMS de otro fabricante.
- Que los resultados de la actividad de los usuarios en el curso puedan ser registrados por el LMS.

Como se puede ver en la siguiente figura, los distintos estándares que se desarrollan hoy en día para la industria del *e-learning* se pueden clasificar en los siguientes tipos:

**1 Sobre el Contenido o Curso:**

Estructuras de los contenidos, empaquetamiento de contenidos, seguimiento de los resultados.

**2 Sobre el Alumno:**

Almacenamiento e intercambio de información del alumno, competencias (habilidades) del alumno, privacidad y seguridad.

**3 Sobre la interoperabilidad:**

Integración de componentes del LMS, interoperabilidad entre múltiples LMS.

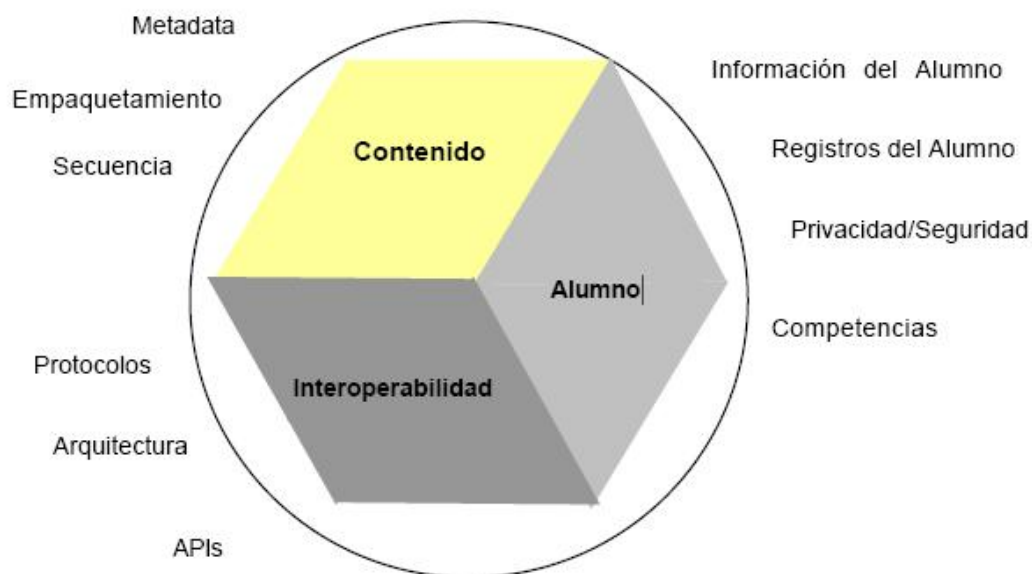


Figura 2.2. Tipología de los estándares *e-learning* [5]

Al hablar sobre un estándar *e-learning*, nos estamos refiriendo a un conjunto de reglas en común para las compañías dedicadas a la tecnología *e-learning*. Estas reglas especifican cómo los fabricantes pueden construir cursos on-line y las plataformas sobre las cuales son impartidos estos cursos de tal manera de que puedan interactuar unas con otras. Estas reglas proveen modelos comunes de información para cursos *e-learning* y plataformas LMS, que básicamente permiten a los sistemas y a los cursos compartir datos o "hablar" con otros.

Esto también nos da la posibilidad de incorporar contenidos de distintos proveedores en un solo programa de estudios.

Estas reglas además, definen un modelo de empaquetamiento estándar para los contenidos. Los contenidos pueden ser empaquetados como "objetos de aprendizaje" (*learning objects* o LO), de tal forma de permitir a los desarrolladores crear contenidos que puedan ser fácilmente reutilizados e integrados en distintos cursos.

Finalmente, los estándares permiten crear tecnologías de aprendizaje más poderosas, y "personalizar" el aprendizaje basándose en las necesidades individuales de los alumnos.

Básicamente, lo que se persigue con la aplicación de un estándar para el *e-learning* es lo siguiente:

- **Durabilidad:** Que la tecnología desarrollada con el estándar evite la obsolescencia de los cursos.
- **Interoperabilidad:** Que se pueda intercambiar información a través de una amplia variedad de LMS.
- **Accesibilidad:** Que se permita un seguimiento del comportamiento de los alumnos.
- **Reusabilidad:** Que los distintos cursos y objetos de aprendizaje puedan ser reutilizados con diferentes herramientas y en distintas plataformas.

Esta compatibilidad ofrece muchas ventajas a los consumidores de *e-learning*.

- Garantizan la viabilidad futura de su inversión, impidiendo que sea dependiente de una única tecnología, de modo que en caso de cambiar de LMS la inversión realizada en cursos no se pierde.
- Aumenta la oferta de cursos disponibles en el mercado, reduciendo de este modo los costos de adquisición y evitando costosos desarrollos a medida en muchos casos.
- Posibilita el intercambio y compraventa de cursos, permitiendo incluso que las organizaciones obtengan rendimientos extraordinarios sobre sus inversiones.
- Facilita la aparición de herramientas estándar para la creación de contenidos, de modo que las propias organizaciones puedan desarrollar sus contenidos sin recurrir a especialistas en *e-learning*.

Estrictamente hablando, no existe un único estándar *e-learning* disponible hoy en día. Lo que existe es una serie de grupos y organizaciones que desarrollan especificaciones (protocolos). Hasta la fecha, ninguna de estas especificaciones ha sido formalmente adoptada como estándar en la industria del *e-learning*. Estas especificaciones no dejan de ser recomendaciones, que por el momento la industria trata de seguir.

## 2.3 Iniciativas de Estándares de e-learning

Tener una idea clara del proceso de estandarización para las tecnologías *e-learning* es una tarea compleja, debido al relativamente poco tiempo que se lleva realizando el proceso y a la profusión de grupos, instituciones y consorcios que trabajan en el tema. No obstante el escenario está mejorando, ya que cada vez más se llegan a acuerdos de colaboración entre distintas iniciativas. Por tanto se está cada vez más cerca de una estandarización real y que tenga un impacto efectivo en la industria.

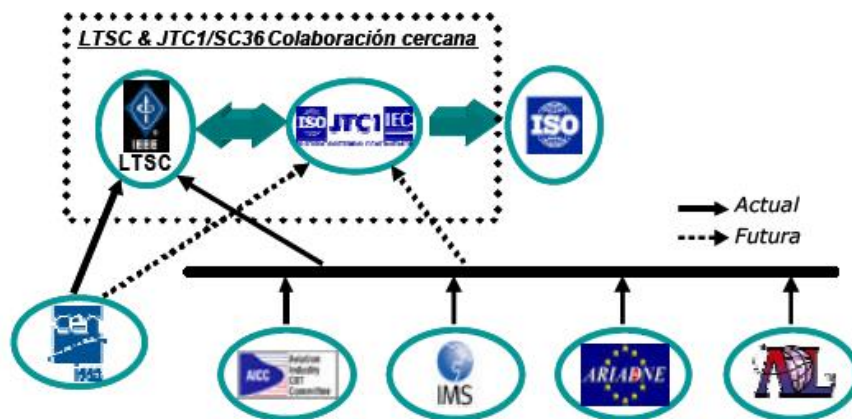


Figura 2.3. Colaboración entre los principales organismos de estandarización [3]

Dentro de las principales iniciativas de estándar para *e-learning* podemos mencionar:

### 2.3.1 AICC, Aviation Industry CBT Commitee

La industria de la aviación ha sido tradicionalmente un gran consumidor de formación, por lo que en 1992 decidieron crear un comité que desarrollase una normativa para sus proveedores de formación basada en computador. De este modo garantizaban la armonización de los requerimientos de los cursos, así como la homogeneización de los resultados obtenidos de los mismos.

Fue el primer organismo creado para crear un conjunto de normas que permitiese el intercambio de cursos CBT (*Computer Based-Training*) entre diferentes sistemas.

Las especificaciones del AICC [10] cubren nueve áreas principales, que van desde los learning objects (*LO*) hasta los learning management systems (*LMS*). Normalmente, cuando una compañía dice que cumple con las especificaciones AICC, significa que cumple con al menos una de estas *guidelines* y recomendaciones (*AICC Guidelines and Recommendations, AGRs*).

La lista completa de *AGRs* es la siguiente [10]:

- AGR 001: AICC Publications
- AGR 002: Courseware Delivery Stations
- AGR 003: Digital Audio
- AGR 004: Operating/Windowing System
- AGR 005: CBT Peripheral Devices
- AGR 006: Computer-Managed Instruction
- AGR 007: Courseware Interchange
- AGR 008: Digital Video
- AGR 009: Icon Standards: User Interface
- AGR 010: Web-Based Computer-Managed Instruction

Aunque la AICC ha publicado varias guías, la más seguida es la AGR 010 que habla de la interoperabilidad de las plataformas de formación y los cursos.

En esta guía se resuelven dos de los problemas fundamentales:

- La carga sin problemas en un LMS de cursos creados por terceros. Este objetivo se consigue definiendo el curso como una entidad totalmente independiente de la plataforma, y creando un sistema (ficheros) de descripción del curso que pueda ser entendido por cualquier plataforma.
- La comunicación entre el LMS y el curso, de tal modo que el curso pueda obtener información necesaria sobre el usuario, y después transmitir los resultados de las interacciones y evaluaciones realizadas por el mismo a la plataforma a fin de su almacenamiento y tratamiento estadístico.

Este segundo objetivo es logrado mediante la definición de un mecanismo de comunicación entre el curso y la plataforma, y un conjunto de datos mínimos que deben ser transmitidos del curso a la plataforma y viceversa. La AICC describe dos mecanismos, uno más sencillo y extendido basado en el protocolo http, y otro mediante una API.

La AICC cuenta con un programa de certificación (a diferencia de las otras iniciativas) y dispone de un *test suite* que le permite a las compañías verificar que sus productos son compatibles con otros sistemas que cumplen con las especificaciones AICC.

Actualmente la AGR 010 de la AICC es el “estándar de facto” en la industria del *e-learning*.

### 2.3.2 IEEE Learning Technologies Standards Committee (LTSC)

Se trata de un organismo que promueve la creación de una norma ISO, una normativa estándar real de amplia aceptación. El LTSC [21] se encarga de

preparar normas técnicas, prácticas y guías recomendadas para el uso informático de componentes y sistemas de educación y de formación, en concreto, los componentes de software, las herramientas, las tecnologías y los métodos de diseño que facilitan su desarrollo, despliegue, mantenimiento e interoperación.

Lo que hizo fue recoger el trabajo del comité de la AICC y mejorarlo, creando la noción de metadata (información sobre los datos, una descripción más detallada que la ofrecida por la AGR 010 de la AICC de los contenidos del curso).

LTSC tiene más de una docena de grupos de trabajo (*working groups o WGs*) y grupos de estudio (*study groups o SGs*) que desarrollan especificaciones para la industria del *e-learning* [21].

Los siguientes grupos de trabajo son parte de las actividades generales de la IEEE LTSC:

IEEE 1484.1 *Architecture and Reference Model*  
IEEE 1484.3 *Glossary*

Los siguientes grupos de trabajo son parte de las actividades relacionadas con los datos y el *metadata*:

IEEE 1484.12 *Learning Object Metadata*  
IEEE 1484.14 *Semantics and Exchange Bindings*  
IEEE 1484.15 *Data Interchange Protocols*

Los siguientes grupos de trabajo son parte de las actividades relacionadas con los LMS y las aplicaciones:

IEEE 1484.11 *Computer Managed Instruction*  
IEEE 1484.18 *Platforms and Media Profiles*  
IEEE 1484.20 *Competency Definitions*

LTSC también trabaja en forma coordinada con otra iniciativa denominada ISO JTC1 SC36, que es un subcomité formado en forma conjunta por la ISO (*International Standard Organization*) y por la IEC (*International Electrotechnical Commission*), dedicado a la normalización en el ámbito de las Tecnologías de la Información para la formación, educación y aprendizaje.

### 2.3.3 IMS Global Learning Consortium, Inc.

Este Consorcio está formado por miembros provenientes de organizaciones educacionales, empresas públicas y privadas. Su misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje *on-line*.

El trabajo de la IEEE fue recogido por esta corporación privada creada por algunas de las empresas más importantes del sector. Su objetivo fue la creación de un formato que pusiese en práctica las recomendaciones de la IEEE y la AICC.

Lo que se hizo fue definir un tipo de fichero XML para la descripción de los contenidos de los cursos. De tal modo que cualquier LMS pueda, leyendo su fichero de configuración IMSMANIFEST.XML, cargar el curso [5].

A continuación se describen algunas de las principales iniciativas de esta organización [16]:

- **Learning Object Metadata (*LOM*)**

Esta especificación entrega una guía sobre cómo los contenidos deben ser identificados o “etiquetados” y sobre cómo se debe organizar la información de los alumnos de manera de que se puedan intercambiar entre los distintos servicios involucrados en un sistema de gestión de aprendizaje (*LMS*). La especificación para metadata del IMS consta de tres documentos: IMS Learning Resource Meta-data Information Model, IMS Learning Resource XML Binding Specifications, IMS Learning Resource Meta-data Best Practices and Implementation Guide.

- **Empaquetamiento de Contenidos (*Content Packaging*)**

Esta especificación [15] provee la funcionalidad para describir y empaquetar material de aprendizaje, ya sea un curso individual o una colección de cursos, en paquetes portables e interoperables. El empaquetamiento de contenidos está vinculado a la descripción, estructura, y ubicación de los materiales de aprendizaje on-line, y a la definición de algunos tipos particulares de contenidos.

La idea es que el contenido desarrollado bajo este estándar sea utilizado en una variedad de sistemas de gestión de aprendizaje (*LMS*). Esta especificación ha sido comercializada por Microsoft bajo el nombre de LRN (*Learning Resource Interchange*).

- **Interoperabilidad de Preguntas y Tests (*Question and Test Interoperability, QTI*)**

El IMS QTI [18] propone una estructura de datos XML para codificar preguntas y test online. El objetivo de esta especificación es permitir el intercambio de estos tests y datos de evaluación entre distintos LMS. Y es la especificación que se usa en este trabajo, por lo que se profundizará en ella más adelante.



- **Empaquetamiento de Información del Alumno (*Learner Information Packaging, LIP*)**

Esta especificación define estructuras XML para el intercambio de información de los alumnos entre sistemas de gestión de aprendizaje, sistemas de recursos humanos, sistemas de gestión del conocimiento, y cualquier otro sistema utilizado en el proceso de aprendizaje. Actualmente, existen varios desarrolladores de productos que tienen en vista adoptar esta especificación.

- **Secuencia Simple (*Simple Sequencing*)**

Esta especificación define reglas que describen el flujo de instrucciones a través del contenido según el resultado de las interacciones de un alumno con el contenido. Esta representación de flujo condicionado puede ser creada manualmente o a través de herramientas compatibles con esta especificación. Una vez creado, la representación de la secuencia puede ser intercambiada entre sistemas diseñados para entregar componentes instruccionales a los alumnos.

- **Diseño del Aprendizaje (*Learning Design*)**

Este grupo de trabajo del IMS investiga sobre las maneras de describir y codificar las metodologías de aprendizaje incorporadas en una solución *e-learning*.

La especificación *IMS Learning Design* (IMS LD) se centra en la descripción de cualquier estrategia didáctica que puede ser aplicada a la estructura organizativa de planes de aprendizaje, cursos o actividades de aprendizaje, facilitando su disponibilidad y ejecución on-line en forma de unidades de aprendizaje (UoL). Un amplio espectro de modelos o estrategias pedagógicas puede ser representado a través de IMS LD, permitiendo a los profesores adaptar de manera flexible sus recursos y escenarios educativos al aprendizaje virtual o mixto (también conocido como blended, es decir, una mezcla del aprendizaje virtual con el presencial), más allá de la secuenciación de actividades y la utilización de repositorios de objetos de aprendizaje.

IMS LD permite la creación de materiales dinámicos y personalizables a través de la descripción actividades para diferentes roles, sus entornos, métodos de ejecución (nivel A), propiedades y condiciones particulares (nivel B), además de notificaciones (nivel C). Por otra parte, facilita la representación y codificación de escenarios para actividades no sólo de naturaleza individual, sino también colaborativas, además de proporcionar los medios para favorecer la comunicación entre los participantes del proceso educativo: docentes y discentes, y entre los propios estudiantes.

El modelo de información de IMS LD utiliza XML y las unidades de aprendizaje expresadas de acuerdo con dicho modelo, pueden ser leídas, procesadas y ejecutadas por un motor que permite su visualización y despliegue. Este motor o IMS LD player es responsable de la coordinación de las actividades e interacciones de los estudiantes y profesores, de mostrar los recursos y del

control del comportamiento de la unidad de aprendizaje de acuerdo con la descripción de su fichero manifiesto.

- **Repositorios Digitales (*Digital Repositories*)**

El IMS está en el proceso de creación de especificaciones y recomendaciones para la interoperación entre repositorios digitales.

La especificación IMS Digital Repositories fue publicada en 2003 con el propósito de facilitar recomendaciones sobre la interoperabilidad de las funcionalidades más comunes de los repositorios. Tales recomendaciones deben ser implementadas a través de servicios que permitan presentar las funcionalidades elegidas con una interfaz de acceso común.

En su nivel más amplio, esta especificación define a los repositorios digitales como cualquier colección de recursos que son accesibles a través de la red sin necesidad de tener conocimientos previos sobre la estructura de dicha colección. Los repositorios pueden contener los recursos como tal o meta-datos que describan a los recursos. En cualquier caso, los recursos y sus meta-datos no tienen que estar depositados necesariamente en el mismo repositorio. Esta especificación tiene como objeto utilizar los esquemas (definición y formato de datos) definidos por otras especificaciones como IMS Meta-Data [17] y Content Packaging [15] en lugar de introducir nuevos esquemas.

- **Definición de competencias (*Competency Definitions*)**

El IMS (al igual que la IEEE en el grupo de trabajo 1484.20) está en el proceso de desarrollar una manera estandarizada de describir, referenciar e intercambiar definiciones de competencias. En esta especificación, el término competencia es usado en un sentido muy general, que incluye habilidades, conocimiento, tareas, y resultados de aprendizaje. Esta especificación entrega una manera de representar formalmente las características principales de una competencia, independiente de su uso en un contexto en particular, permitiendo así su interoperabilidad entre distintos LMS.

- **Accesibilidad (*Accessibility*)**

Este grupo de trabajo promueve el contenido de aprendizaje accesible a través de recomendaciones, guías y modificaciones a otras especificaciones. Tecnología accesible se refiere a la tecnología que puede ser usada sin tener acceso pleno a uno o más canales de entrada y salida para la representación de la información, usualmente visuales y auditivos.

#### 2.3.4 ADL SCORM

Formada en 1997, la iniciativa ADL (*Advanced Distributed Learning*) [9], es un programa del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca para desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y en gran escala, de formación educativa sobre nuevas tecnologías Web.

Este organismo recogió “lo mejor” de las anteriores iniciativas (el sistema de descripción de cursos en XML de la IMS, y el mecanismo de intercambio de información mediante una API de la AICC) y las refundió y mejoró en su propio estándar: SCORM, *Shareable Content Object Reference Model* (Modelo de Referencia para Objetos de Contenidos Intercambiables).

SCORM proporciona un marco de trabajo y una referencia de implementación detallada que permite a los contenidos y a los sistemas usar SCORM para “hablar” con otros sistemas, logrando así interoperabilidad, reusabilidad y adaptabilidad.

Todo esto se reafirma mediante las siguientes posibilidades:

- La disponibilidad de un Sistema de Gestión de Aprendizaje o LMS basado en Web para lanzar diferentes contenidos que se han desarrollado por varios autores usando herramientas de diversos vendedores.
- La disponibilidad de diversos LMS producidos por diferentes vendedores para lanzar un mismo contenido.
- Y la disponibilidad de múltiples productos o entornos LMS basados en Web para acceder a un repositorio común de contenidos

Las especificaciones de SCORM están organizadas como “libros” separados. La mayoría de estas especificaciones son tomadas desde otras organizaciones. Estos “libros” técnicos se agrupan bajo dos tópicos principales: *Content Aggregation Model* y *Run-Time Environment*.

Como lo muestra la siguiente figura, la actual versión 1.2 de SCORM ha sido dividida en tres libros que se detallan a continuación:

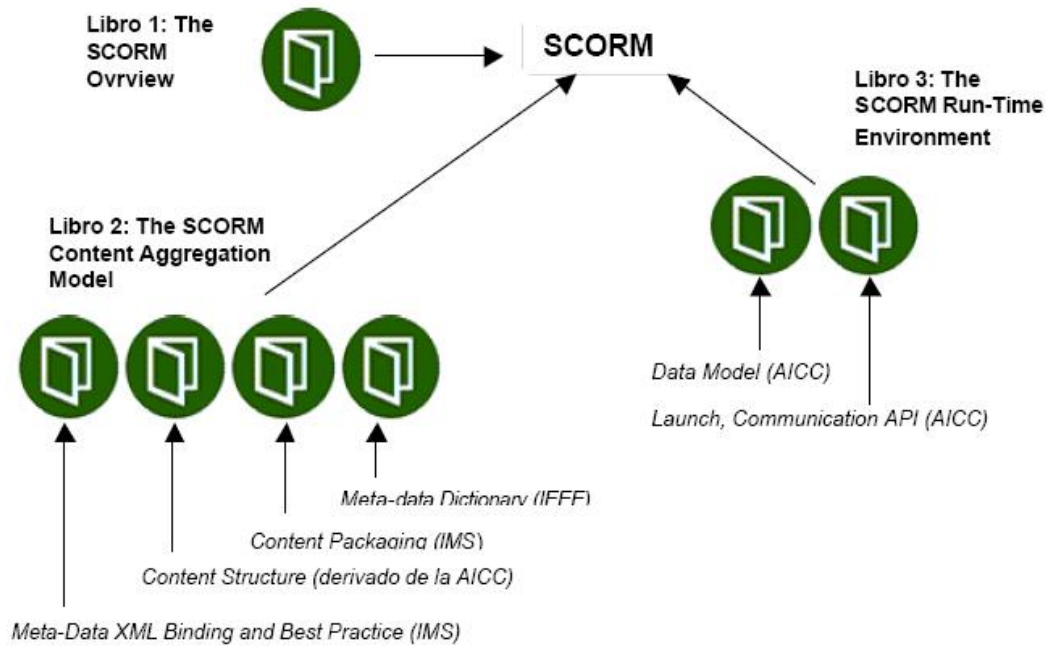


Figura 2.4. División en libros de la actual versión de SCORM [5]

- **Libro 1: *Scorm Overview*.** Contiene una descripción general de la iniciativa de ADL, un análisis de SCORM, y un resumen de las especificaciones técnicas contenidas en las siguientes secciones.
- **Libro 2: *Scorm Content Aggregation Model*.** Contiene una guía para identificar y agregar recursos dentro de un contenido de aprendizaje estructurado. Este libro describe una nomenclatura para el contenido de aprendizaje, describe el *SCORM Content Packaging* (empaquetamiento de contenidos) y hace referencia al *IMS Learning Resource Meta-data Information Model*, el cual está basado en el *IEEE LTSC Learning Object Metadata (LOM) Specification*, que fue el resultado de un esfuerzo en conjunto entre el *IMS Global Learning Consortium* y la *Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE)* [12].
- **Libro 3: *Scorm Run-Time Environment*.** Incluye una guía para lanzar contenidos y hacerle un seguimiento en un ambiente basado en Web. Este libro es derivado del *CM1001 Guidelines for Interoperability* de la AICC.

SCORM también divide el mundo de la tecnología *e-learning* en componentes funcionales. Los principales componentes son: Learning Management System (LMS) y Sharable ContentObjects (SCOs). SCO se refiere a objetos de aprendizaje reusables y estandarizados. Otros componentes en el modelo SCORM son herramientas que crean los SCOs y los ensamblan en unidades de aprendizaje más grandes (un curso por ejemplo).

- **SCORM Content Aggregation Model**

El objetivo del modelo de agregación de contenidos de SCORM es proveer un medio común de componer contenidos educativos desde diversas fuentes compartibles y reusables.

Define cómo un contenido educativo puede ser identificado, descrito y agregado dentro de un curso o una parte de un curso, y cómo puede ser compartido por diversos LMS o por diversos repositorios.

El modelo, incluye especificaciones para los **metadata** y el **CSF** (*Content Structure Format*):

- Los *metadata* (datos sobre los datos) constituyen la clave para la reusabilidad. Describen e identifican los contenidos educativos, de manera que pueden formar la base de los repositorios. Se han especificados basándose en las recomendaciones *IEEE LSTC Learning Object Metadata (LOM)*. Los metadata se aplican a tres niveles: a los "assets" (elementos de contenidos de más bajo nivel), a los SCO (*Shareable Content Objects*) y bloques de SCO's, y al CSF.
- *Content Structure Format*. El proceso de diseño y creación de un curso comprende la construcción de un conjunto de objetos de contenidos educativos, relacionados entre sí mediante cierta estructura. Este es el objetivo del Content Structure Format (*Formato para la Estructura de los Contenidos*), proporcionar un medio de agregación de bloques de contenidos, aplicando una estructura y asociándola a una taxonomía para que tengan una representación y un comportamiento común en cualquier LMS.

El modelo CSF ha sido desarrollado a partir de las especificaciones CMI (*Computer Managed Instruction*) de la AICC. Posteriormente, realizando una reorganización entre las especificaciones de ADL, el AICC, el IEEE e IMS Global Learning Consortium, se ha llegado aun nuevo modelo representado en la *IMS Content Packaging Specification*, tal y como se expone a continuación.

Un CSF es un componente necesario para mover un contenido educativo de un lugar a otro, pero no es suficiente por sí mismo. Es necesario agregar y guardar los contenidos en un paquete. Para ello está diseñado el *Content Packaging*. *Packaging* o empaquetar, es el proceso de identificar todos los recursos necesarios para representar los contenidos educativos y después reunir todos los recursos junto a un manifiesto.

ADL señala en su nueva versión que el CSF de SCORM V1.1 no es adecuado para el empaquetamiento, y por lo tanto sus elementos han sido "remapeados" en dos nuevas estructuras, *Content Aggregation Package Application Profile* y *Content Aggregation Manifest*, que

incluyen la mayoría de la información del anterior CSF pero que añade un nuevo método de inventariar todos los ficheros requeridos para distribuir los contenidos e identificar sus relaciones.

- **SCORM Run-Time Environment.**

El objetivo del entorno de ejecución de SCORM es proporcionar un medio para la interoperabilidad e interacción entre los objetos compartibles de contenidos (SCO), y los sistemas de gestión de aprendizaje (LMS).

Un requerimiento de SCORM es que el contenido educativo sea interoperable a través de múltiples LMS, sin tener en cuenta las herramientas que se usen para crear o usar los contenidos.

Para que esto sea posible, debe existir un método común para lanzar un contenido, un método común para que los contenidos se comuniquen con el LMS y elementos de datos predefinidos que sean intercambiables entre el LMS y el contenido durante su ejecución.

Los tres componentes del entorno de ejecución de SCORM son:

1. El *launch* o despliegue. Es el mecanismo que define el método común para que los LMS lancen un SCO basado en Web. Este mecanismo define los procedimientos y las responsabilidades para el establecimiento de la comunicación entre el contenido a mostrar y el LMS. El protocolo de comunicación está estandarizado a través del uso común del API. El LMS puede implementar la presentación de los SCO del modo que se desee, por ejemplo, desarrollando un mecanismo de adaptación al usuario (mediante técnicas de aprendizaje simbólico), o bien, puede delegar esa responsabilidad al cliente permitiéndole que navegue por el curso libremente a través de menús.
2. La API (*Application Program Interface*). Es el mecanismo para informar al LMS del estado del contenido (por ejemplo si está inicializado, finalizado o en error) y es usado para intercambiar datos entre el LMS y los SCO (por ejemplo datos de tiempo, de puntuación, etc.). La API es simplemente un conjunto de funciones predefinidas que se ponen a disposición de los SCO, como por ejemplo *LMSInitialize* o *LMSSetValue*.
3. El Modelo de Datos. Es una lista estandarizada de elementos usados para definir la información a intercambiar, por ejemplo, el estado del contenido educativo. Son elementos de datos que tanto el LMS como el SCO van a conocer como el umbral de terminación para saber cuando un estudiante ha alcanzado las competencias que se necesitan obtener. Es responsabilidad del LMS mantener el estado de los datos requeridos a lo largo de las sesiones, y el SCO los utilizará en el caso de que se necesite su reutilización entre una sesión y otra.

Existen diversos modelos de datos en desarrollo en varias organizaciones. Incluyen datos sobre el perfil del estudiante, información de estado, iteraciones sobre test y preguntas, evaluaciones, etc. El Modelo SCORM está basado directamente del CMI Data Model del AICC.

## *2.4 Plataformas de enseñanza (LMS)*

En español se ha denominado plataforma de aprendizaje o LMS (Learning Management System) a un programa informático que se utiliza para la creación, gestión y distribución de actividades formativas a través de la web. Son aplicaciones que facilitan la creación de entornos de enseñanza y aprendizaje, integrando materiales didácticos y herramientas de comunicación, colaboración y gestión educativa.

Las plataformas ofrecen ambientes de aprendizaje ya diseñados e integrados. A ellos acceden los alumnos a través de una clave personal. Por ello, se trata de un espacio privado, dotado de las herramientas necesarias para aprender (comunicación, documentación, contenidos, interacción, etcétera). Además, las plataformas permiten hacer un mejor seguimiento del progreso de los alumnos.

Es una aplicación que usualmente reside en un servidor web en la que se desarrollan las acciones formativas. Es el lugar donde alumnos, tutores, profesores o coordinadores se conectan a través de Internet para descargarse contenidos, ver el programa de asignaturas, realizar actividades formativas, enviar un correo electrónico al profesor, charlar con los compañeros, debatir en un foro o participar en una tutoría.

Asimismo, todo LMS consta de un entorno de aprendizaje y comunicación, al que acceden los alumnos y profesores y un entorno de administración, desde dónde se configuran los cursos, se dan de alta los alumnos, se importan contenidos, se habilitan servicios, etcétera.

Mediante distintos tipos de herramientas, el LMS permite entre otras cosas que:

- Los profesores coloquen a disposición de los alumnos los objetivos del curso, su contenido y su reglamentación.
- Los tutores y coordinadores supervisen el desarrollo del curso y el avance de cada alumno.
- Los alumnos accedan a los contenidos, realicen las asignaciones previstas, se comuniquen entre sí y el con el tutor para resolver dudas y realizar trabajos en grupo.

- Los administradores obtengan información on-line del progreso del curso y de las acciones administrativas relacionadas, tales como inscripción de alumnos, historial de cursos, etc.

Para poder cumplir con su propósito el LMS posee un conjunto mínimo de herramientas que pueden agruparse de la siguiente forma:

**Herramientas de creación y distribución de contenidos:** editor de contenidos on-line, repositorios de archivos de imágenes, de vídeo y de texto como biblioteca on-line, sistema de reconocimiento de contenidos de CD, inserción de hipervínculos, imágenes y vídeos, administración de calendario de contenidos.

**Herramientas de comunicación y colaboración:** foros de discusión, chats, formación de grupos de trabajo, comunicación con el autor, miembros del curso, novedades y calendario del curso.

**Herramientas de seguimiento y evaluación:** estadísticas y ficha personal por alumno, seguimiento de cada actividad, sistemas de exámenes editables por el docente o tutor, reportes de actividad.

**Herramientas de administración y asignación de permisos:** concesión de permisos y autorizaciones, asignación de permisos por perfil de usuario, administración personal de perfiles de usuario, proceso de inscripción, planes de carrera y oferta formativa.

En estos momentos, en el mercado existe una gran variedad de plataformas LMS, algunas de ellas pertenecientes al mundo del software libre y otras de pago. Algunos ejemplos de este tipo de software para la enseñanza on-line se presentan en las siguientes secciones.

#### 2.4.1 Moodle

Moodle [22] es una plataforma LMS de código abierto que fue creada por Martin Dougiamas, quien trabajó como administrado de WebCT, otra importante plataforma LMS, en la Universidad de Curtin. Esta basada en trabajos sobre el constructivismo en pedagogía, que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de los libros y explicaciones del profesor.

Esta herramienta ha evolucionado desde 1999, actualmente su última versión disponible es la 2. Moodle tiene una gran comunidad de usuarios con aproximadamente 150.000 usuarios registrados, que hablan 75 lenguas diferentes y provienen de 160 países.

La palabra Moodle era al principio un acrónimo de Modular Object-Oriented Dynamic Learning Environment, lo que resulta fundamentalmente útil para programadores y teóricos de la educación. También es un verbo que describe



el proceso de deambular perezosamente a través de algo, y hacer las cosas cuando se te ocurre hacerlas. Las dos acepciones se aplican a la manera en que se desarrolló Moodle y a la manera en que un estudiante o profesor podría aproximarse al estudio o enseñanza de un curso on-line.

Moodle es fácil de instalar en cualquier PC, sus requisitos de instalación, tanto software como hardware son asequibles. Es un LMS sencillo, ligero, eficiente y con una presentación compatible con cualquier navegador web.



Figura 2.5. Página de inicio de Moodle [22]

En cuanto a la gestión de usuarios, esta plataforma tiene las siguientes funcionalidades:

- **Inscripción:** El sistema asigna una clave de acceso a cada usuario para evitar el acceso a personas ajenas a los cursos, permite la inscripción y eliminación de alumnos manualmente por parte de los profesores y cada cuenta de usuario puede recibir diferentes permisos.
- **Perfiles:** El sistema permite crear distintos perfiles de usuario para cada curso. El perfil administrador controla la creación de cursos y profesores añadiendo personal al curso. Los profesores pueden, a su vez, tener varios perfiles y permisos de forma que puedan o no editar cursos, añadir alumnos y el perfil de estudiante, que permite visualizar los contenidos y participar en las actividades planificadas por los profesores o tutores.

El objetivo del módulo de gestión de usuarios es reducir al mínimo la importancia del administrador, pero manteniendo la seguridad. Soporta varios mecanismos de autenticación así como la validación mediante LDAP. Los estudiantes son animados a crear un perfil on-line con fotos, descripción, etcétera y la interfaz web de cada usuario puede configurarse para varios idiomas y zonas horarias.

En cuanto a la gestión del curso, Moodle presenta un gran conjunto de funcionalidades divididas en módulos:

- **Módulo de asignaciones:** Este módulo permite añadir asignaciones a un curso, con una fecha de vencimiento. Las entregas fuera de plazo están permitidas y muestra al profesor el tiempo transcurrido desde la finalización del plazo hasta la entrega. Los alumnos tienen un interfaz para la entrega de ficheros. Además los profesores pueden añadir comentarios y una evaluación a cada estudiante.
- **Módulo de chat:** Este módulo permite la comunicación síncrona entre miembros del curso incluyendo la imagen de cada participante en la ventana de diálogo. Soporta formato HTML, imágenes e iconos. Todas las sesiones pueden ser guardadas para quedar accesibles con posterioridad tanto para profesores como para alumnos.
- **Módulo de foro:** Permite la comunicación entre los distintos roles del curso de forma asíncrona. Existen diferentes tipos de foros, solo para profesores, foros de noticias, foros abiertos,... Cada mensaje publicado en el foro lleva adjunta la foto del autor y las discusiones pueden ser visualizadas de forma llana o anidada, con los más antiguos o los más nuevos primero. Los profesores pueden no permitir las respuestas a un mensaje, por ejemplo en un foro de noticias. Además los foros permiten la suscripción, enviando por correo electrónico las novedades a los participantes.
- **Módulo de glosario:** Este módulo permite al profesor crear un glosario con los términos más importantes de un curso.
- **Módulo de lección:** Una lección es un conjunto de páginas en las cuales se presenta un tema. Se soportan un gran número de formatos, entre ellos SCORM. Además puede contener ciertas preguntas que sirven para evaluar el grado de asimilación del tema presentado por parte de los alumnos, estas preguntas pueden ser de varios tipos como respuesta múltiple, respuesta corta, etcétera.
- **Módulo de FAQ:** Los profesores pueden crear una batería de preguntas para ser reutilizadas en diferentes FAQs, éstas pueden ser archivadas por categorías para facilitar su acceso. Las preguntas permiten el formato HTML y añadir imágenes y pueden ser importados ficheros de texto externos. El profesor puede definir una fecha, superada la cual los alumnos no podrán acceder al FAQ.
- **Módulo de recursos:** Este módulo permite almacenar ficheros, tales como presentaciones en PowerPoint, videos, textos, etcétera. Pueden almacenarse estos contenidos de forma local o remota y también pueden ser igualmente gestionados. Es posible crear carpetas para organizar su contenido. Además el módulo dispone de un editor de archivos HTML y de un reproductor MP3.

- **Módulo de consulta:** Se proporcionan encuestas, colección de tests que pueden configurarse en el momento o importar algunos ya preparados y contrastados. Además incluyen instrumentos para el análisis de clases on-line. Los resultados están siempre disponibles, incluyendo gráficos y también pueden descargarse en formato Excel o en texto plano. A cada estudiante se le informa de sus resultados, comparados en el promedio de la clase.
- **Módulo de Wiki:** Permite la creación de una página Wiki sobre el curso, en la que los participantes puedan añadir sus comentarios.
- **Módulo de taller:** Este módulo permite la evaluación de contenidos entre estudiantes y el profesor puede actuar revisando la evaluación.

Además de estos módulos, la gestión del curso de Moodle permite al profesor administrador controlar toda la configuración y elegir la interfaz del curso y su estructura, por semanas, por temas o basado en discusión. Pueden definirse grupos a los cuales pueden pertenecer profesores y alumnos, incluso un mismo individuo puede pertenecer a más de un grupo. Además Moodle posee la capacidad de crear un archivo comprimido (ZIP) con toda la información relativa al curso: contenidos y su configuración como backup. También posee un calendario para los cursos, de esta manera los profesores pueden añadir a este calendario citas, anuncios o eventos. Así, puede ser restituido en el servidor en caso de pérdida de datos.

Moodle es actualmente el LMS de código abierto más implantado en todo el mundo. La Figura 2.5 muestra la página inicial de Moodle.

#### 2.4.2 .LRN

.LRN es una plataforma de software libre para comunidades de aprendizaje e investigación. Desarrollado inicialmente por el MIT como parte del Intellectual Commons, .LRN está respaldado en la actualidad por un consorcio mundial de instituciones educativas, organizaciones sin ánimo de lucro, empresas y desarrolladores de código abierto [20].

.LRN tiene capacidades de gestión de cursos, comunidades on-line, gestión de contenidos y del aprendizaje. La versión actual de .LRN es la 2.3.1, está implementada en Java y contiene un gran número de funcionalidades. Algunos de los módulos más importantes son:

- **Evaluaciones:** Este módulo permite la evaluación del desempeño de los estudiantes. Es posible crear varios tipos de evaluaciones, por ejemplo, revisiones, exámenes, etcétera y permite al profesor fijar el tiempo requerido para realizar la prueba. Además, permite reutilizar contenidos ya sean preguntas, cuestiones, conjuntos de soluciones múltiples, etcétera. Este módulo es acorde a la especificación LMS QTI,

por tanto, permite importar y exportar evaluaciones usando un fichero zip.

- Correo electrónico: .LRN asume que todos los usuarios antes de registrarse tienen una dirección de correo electrónico. .LRN proporciona una manera sencilla de integrar un sistema de correo existente en una organización con la plataforma educativa. Además, es posible utilizar un módulo que permite el envío de correos electrónicos a múltiples usuarios al mismo tiempo, por ejemplo a todos los miembros de un grupo, definiendo la hora del envío y el formato de texto, HTML o plano. También pueden enviarse correos automáticamente a aquellos alumnos que, por ejemplo, no hayan entregado una asignación o no hayan obtenido una evaluación positiva en un examen.
- Calendario: .LRN posee la capacidad de dotar a los cursos de un calendario. Los profesores pueden añadir a este calendario citas, anuncios o eventos sobre la planificación del curso. Los alumnos también pueden añadir sus citas personales a este calendario y consultarlo en varias vistas, citas del día, de la semana, del mes. Los subgrupos creados en el curso, también pueden tener su propio calendario.
- FAQ: .LRN permite al profesor de un curso elaborar listas o bloques de preguntas frecuentes que estarán a disposición de los alumnos para su consulta.
- Almacén de ficheros: Cualquier usuario dispone de un espacio privado en el que depositar sus ficheros personales. Éstos pueden ser compartidos por varios usuarios. Los subgrupos creados pueden mantener un área de intercambio de ficheros. Además, mantiene un sistema de versiones, de forma que la subida al almacén de una nueva versión de un fichero, no borra la antigua.
- Foro: Esta plataforma soporta la creación de discusiones en foros. Estos foros pueden ser anidados o llanos, moderado o sin moderar, de participación abierta, cerrada; por tanto, son foros altamente flexibles. Soportan una integración con el correo electrónico para enviar mensajes a los participantes como notificaciones de cambios y suscripciones o para enviar mensajes al foro utilizando el e-mail. Los subgrupos pueden tener sus propios foros.
- Noticias: Permite, de forma fácil, la inclusión de noticias o alumnos en un panel virtual del curso. Es un mecanismo de comunicación unidireccional desde el profesor hacia los alumnos. Al incluir una nueva noticia, el profesor de un curso puede definir un periodo, superado el cual la noticia deja de aparecer en el tablón.
- Álbum de fotos: Esta funcionalidad permite a los alumnos subir al servidor fotos y compartirlas con el resto de usuarios. Su

funcionamiento es muy similar a del almacén de ficheros, pero está dirigido a compartir fotografías.

- Programa del curso: Esta funcionalidad permite al profesor publicar el programa del curso para que sea accesible por los alumnos.
- Lista de profesores: Muestra la información relativa al conjunto de profesores, instructores o tutores que formar la plantilla del curso. Muestra información de contacto y otros datos de cada uno de ellos.
- Weblog: .LRN se distribuye con un paquete que permite a los alumnos disponer de su propio espacio en el cual colocar sus comentarios. Soporta RSS.
- Encuestas: Permite la realización de encuestas entre los usuarios del curso. Pueden ser útiles para averiguar la opinión de los alumnos sobre algún tema, o fijar eventos de planificación. Los datos obtenidos pueden ser exportados en formato Excel. En la siguiente versión de la plataforma se soportarán las encuestas anónimas.

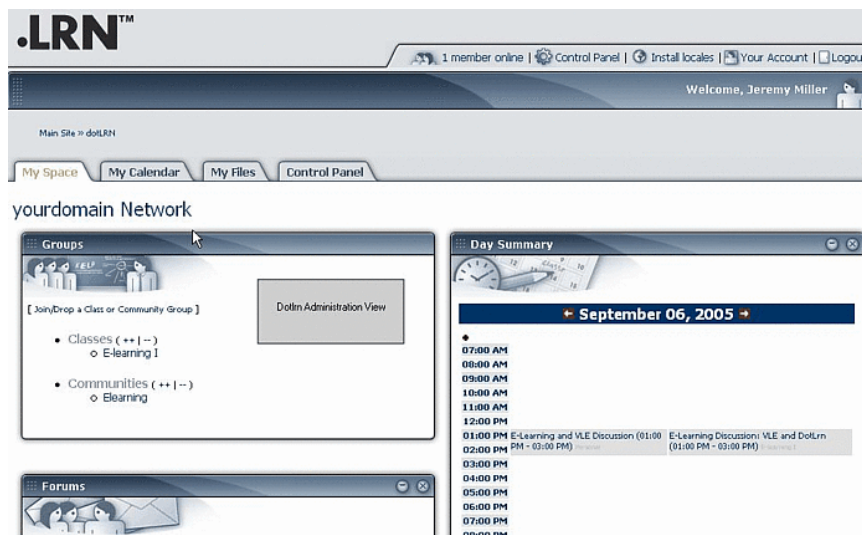


Figura 2.6. Página de administración de .LRN [20]

.LRN soporta diferentes lenguas y zonas horarias, constantemente se está actualizando para soportar nuevos idiomas. La Figura 2.6 presenta la página de administración de .LRN.

### 2.4.3 BlackBoard/WebCT

WebCT es una plataforma comercial que fue originalmente desarrollado en la Universidad de Columbia Británica, en Canadá por un miembro de la facultad de Ciencias en Computadores que en 1995 comenzó la búsqueda de sistemas

basados en páginas web desarrolladas para la educación. Su investigación demostró que la satisfacción del estudiante y el desempeño académico podían mejorar a través del uso de recursos educativos basados en páginas web. Para continuar su investigación decidió construir un sistema que facilitara la creación de entornos educativos basados en esta tecnología. Así se originó la primera versión de WebCT [26]. En 1997, se creó la compañía Corporación de Tecnologías Educativas WebCT. Hacia 1999 tenía alrededor de 2 a 3 millones de estudiantes en 30 países. En ese mismo año, WebCT fue adquirido por ULT (Universal Learning Technology), una empresa con sede en Boston.

El 28 de febrero de 2006, WebCT y Blackboard, otra empresa que desarrolla entre otros productos, software para la educación a distancia, anunciaron su fusión. Se acordó que la compañía creada llevara el nombre de Blackboard. La última versión de su plataforma LMS es Blackboard Learning System.

La Figura 2.7 muestra la página de inicio de la plataforma BlackBoard.

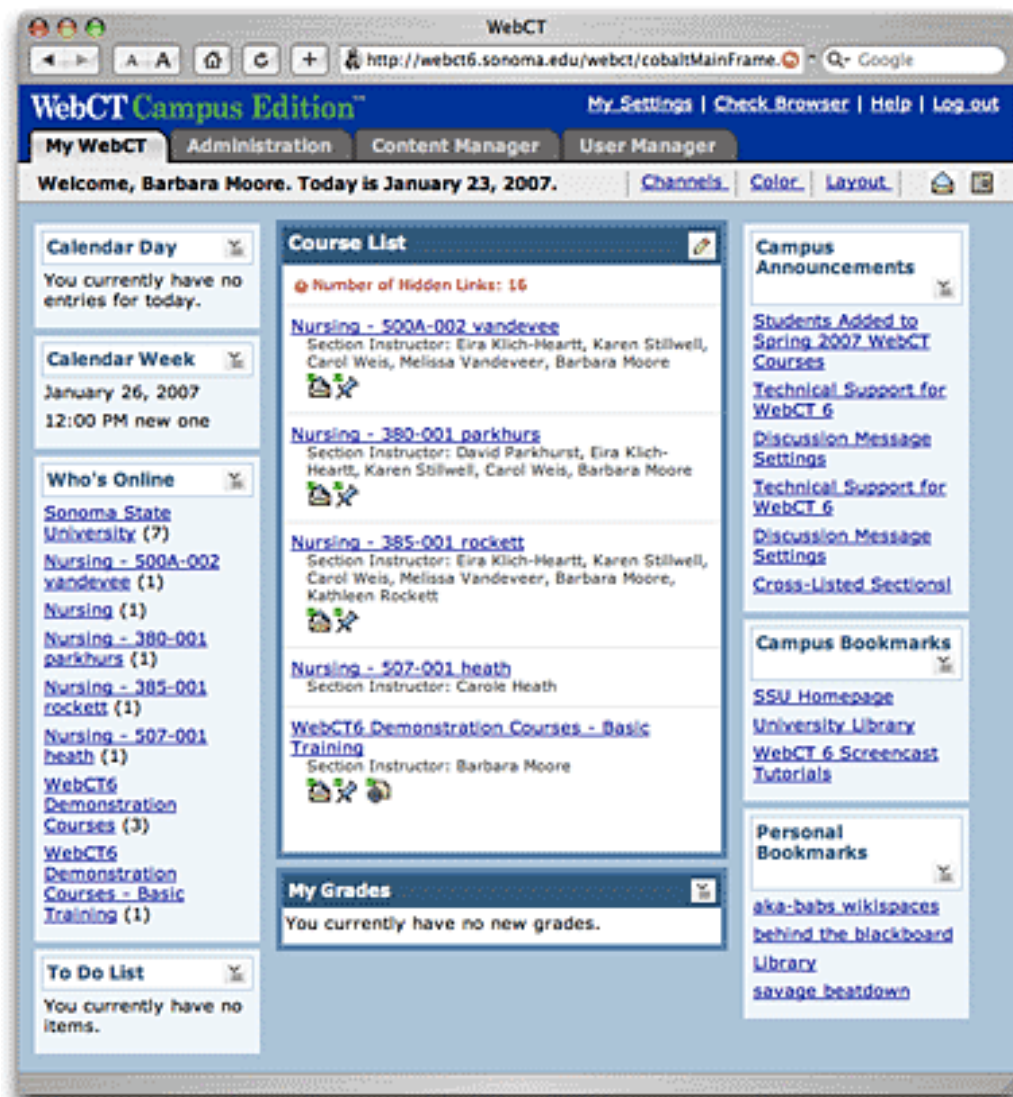


Figura 2.7. Página de inicio de la plataforma BlackBoard [26]

La herramienta Blackboard Learning System ofrece un conjunto de funcionalidades de fácil uso. Están divididas en tres áreas, instrucción, comunicación y evaluación.

**Instrucción.** Este conjunto de funcionalidades está pensado para favorecer la creación y la gestión de los cursos. Algunas de estas funcionalidades son:

- **Gestión de cursos:** Permite una fácil y eficaz creación de nuevos cursos así como su instalación en la plataforma. También provee la capacidad de realizar migraciones semestrales.
- **Autoría de contenidos:** Permite la creación y edición de contenidos de aprendizaje mediante un sencillo editor.
- **Adaptación del curso:** Permite al profesor definir diferentes formas de guiar el curso estableciendo cuándo los alumnos pueden acceder a cada lección, discusión, tarea, o cualquier otra actividad de aprendizaje.
- **Programa del curso:** Esta funcionalidad permite al profesor, entre otras cosas, crear glosarios sobre términos relativos a la temática del curso y a mantener información acerca de los profesores o instructores que participan en el curso.
- **Gestión de información personal:** Está dirigida a cualquier usuario de la plataforma, sea cual sea su perfil. Permite la gestión de la información personal, incluyendo un calendario.

**Comunicación.** Las funcionalidades de comunicación proporcionan a los alumnos y profesores, la capacidad de discutir on-line sobre diversos temas u organizar sesiones de trabajo en grupo sin barreras geográficas. Algunas de las funcionalidades que presenta BlackBoard son:

- **Panel de discusión:** Similar a un foro, permite crear discusiones anidadas y asíncronas. Los profesores pueden crear foros para discutir sobre diversos temas relacionados con diferentes aspectos del curso.
- **Creación de grupos:** Los profesores pueden usar esta funcionalidad para crear grupos de alumnos. Cada grupo puede tener su propio espacio de intercambio de ficheros, su panel de discusión, etcétera.
- **Aula virtual:** Similar a un chat, permite mantener sesiones de charla síncrona basada en el intercambio de mensajes de texto.

**Evaluación.** Las funcionalidades de evaluación dan a los profesores la capacidad de examinar a y evaluar el aprendizaje de los alumnos. Estas funcionalidades aumentan la eficiencia del profesor en la tarea de evaluar a los estudiantes. Estas funcionalidades son:

- **Evaluaciones:** Permite a los profesores crear varios tipos de evaluaciones, como por ejemplo exámenes, para la evaluación de los

alumnos. Estos exámenes son automáticamente corregidos y valorados. Soporta varios tipos de evaluaciones, preguntas de verdadero o falso, de múltiples respuestas, o rellenar huecos, etcétera.

- Asignaciones: Ayuda a los profesores en la creación de tareas para los alumnos. Éstos pueden enviar sus respuestas a través de la plataforma.
- Libro de notas: Almacena los resultados que los alumnos van obteniendo a lo largo del curso.

Además de todas estas funcionalidades y herramientas, BlackBoard Learning System incluye capacidades para escalar la plataforma según el número de alumnos, para dar soporte a varias lenguas, incluso soporta los idiomas chino y japonés. También implementa algunas de las especificaciones del IMS Global, como IMS Content Packaging, IMS Metadata Specification o IMS Question and Test Interoperability.

BlackBoard Learning System es parte del BlackBoard Academic Suite, una familia de productos comerciales dedicados a la formación y al intercambio de conocimientos on-line.

#### 2.4.4. Otras plataformas LMS

Actualmente, existen en el mercado muchas más plataformas LMS, algunas de ellas de código abierto y otras comerciales. Algunos ejemplos son:

- ANGEL Learning: conjunto de herramientas para la enseñanza a través de Internet. Es una plataforma LMS comercial. Su última versión es ANGEL Learning Suite 7.  
<http://www.angellearning.com/products/lms/>
- ATutor: plataforma web para *e-learning* de código abierto que también contiene funciones de creación de contenidos, por tanto, puede considerarse LCMS. Esta plataforma ha adoptado las especificaciones IMS/SCORM Content Packaging.  
<http://www.atutor.ca>
- Claroline: es una plataforma web de libre distribución dedicada a la gestión del aprendizaje on-line. Está basada en PHP y MySQL. Claroline está traducido a 34 idiomas y es usado por cientos de organizaciones en todo el mundo.  
<http://www.claroline.net>



---

### 3. PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN

---

Durante el proceso educativo, los exámenes son utilizados ampliamente como herramienta para evaluar si un alumno ha asimilado los conceptos que le han sido presentados y como herramienta de autoevaluación para el alumno, de manera que éste pueda reforzar aquella parte de la materia para la que no tenga un dominio suficiente. En este sentido, el uso de herramientas informáticas ha supuesto un avance. Un gran número de tipos de preguntas pueden ser corregidas de manera automática mediante el uso de este tipo de herramientas, de manera que un profesor puede crear una batería de preguntas que el sistema informático puede utilizar para preparar exámenes y corregirlos automáticamente.

De hecho, en la actualidad, gran parte de las plataformas de aprendizaje (LMS) incluyen con menor o mayor funcionalidad una herramienta para la creación, gestión y realización de exámenes en línea. Sin embargo, acorde con la idea que se ha venido transmitiendo en la memoria de este trabajo de fin de carrera, el esfuerzo realizado por el instructor en la elaboración de preguntas y exámenes puede llegar a perderse si estos sólo se implantan para un sistema con requisitos específicos y es necesario cambiar de plataforma de aprendizaje.

Asimismo, ya que la creación de dichos exámenes requiere invertir cierto esfuerzo, sería deseable tener la posibilidad de compartir este esfuerzo permitiendo el intercambio de exámenes completos, o poder crear repositorios de preguntas. Así mismo, estas preguntas deberían estar clasificadas por materias y por dificultades para simplificar su localización y reutilización en la formulación de nuevos exámenes. Por lo tanto, este es el problema que trata de resolver este proyecto.

Como ya se ha mencionado previamente, la especificación IMS Question and Test Interoperability (IMS QTI) permite crear preguntas individuales y

evaluaciones completas. El objetivo principal de esta especificación es permitir el intercambio de preguntas, evaluaciones y resultados entre distintas herramientas. Con este propósito IMS QTI plantea un modelo en el que se definen los componentes principales que intervienen en el proceso de evaluación y, adicionalmente a este modelo, se proporciona un formato de contenido para almacenar las preguntas de manera independientemente del sistema o herramienta de autoría utilizada para crearlas.

Este formato, basado en XML, hace uso de estándares ampliamente utilizados en el ámbito empresarial y técnico, permitiendo el uso de las mismas preguntas entre diversos sistemas de gestión de aprendizaje o LMS, entre sistemas de evaluación electrónica independientes y la integración en un único LMS de preguntas y exámenes desarrollados con distintas herramientas. Por otro lado se propone un sistema coherente para que los sistemas puedan informar de cual es el resultado de una evaluación.

IMS QTI permite la construcción de almacenes o repositorios de preguntas que sean directamente utilizables en distintos sistemas LMS (e incluso para crear e imprimir exámenes tipo test que los alumnos realicen por escrito). Esto puede ser muy útil cuando se generalice la representación mediante QTI de los repositorios de libre acceso existentes.

La generalización de este tipo de almacenes y su libre disposición en formato compatible con otras plataformas puede simplificar mucho la tarea de los docentes, especialmente dada la situación actual en la que diversos LMS comerciales o libres dan algún tipo de soporte a este formato.

En el ámbito hispano, muy centrado en los exámenes de desarrollo de conceptos, puede parecer que este tipo de exámenes basados en tests son una forma excesivamente simplista de realizar evaluaciones pero la aceptación en la industria y sobre todo en el mundo anglosajón de este tipo de pruebas es muy grande. De hecho es mediante pruebas, principalmente de tipo test y con corrección automática, con lo que se otorgan calificaciones para saber el nivel de inglés de los alumnos que quieren cursar estudios en universidades americanas o con los que se puede obtener acreditaciones profesionales respaldadas por empresas en el campo informático.

En este trabajo se aborda la implementación de un sistema que permite la búsqueda y creación de tests basado en la solución propuesta por el Método de Desarrollo de Materiales Didácticos (MD2)[6] que permite describir los componentes de un material, en este caso los test para localizar sus componentes y facilitar su creación.

Para la implementación de la aplicación web se ha utilizado el patrón Modelo Vista Controlador. Este patrón es muy utilizado para la creación de aplicaciones web, ya que permite separar en tres capas la lógica del programa. Como lenguaje de programación se ha escogido Java, en primer lugar, por los conocimientos previos que de este lenguaje se poseían, en segundo lugar por la gran versatilidad del lenguaje, que permite tanto la programación en forma de scripts, para la presentación de la información al

usuario, como la programación de la lógica de negocio de la aplicación, en tercer lugar, por la existencia de varias herramientas software de código abierto que existen para este lenguaje, desde entornos de desarrollo hasta contenedores web, y por último, en cuarto lugar, por la existencia del API Struts, que permite implementar fácilmente el patrón Modelo Vista Controlador en este lenguaje. En las secciones siguientes se explican las tecnologías empleadas en la solución propuesta por este trabajo.

### 3.1 Arquitectura de la solución

La arquitectura de la aplicación, como ya se ha dicho, sigue el patrón MVC (Modelo-Vista-Controlador) ya que un problema común a la hora de desarrollar aplicaciones, es el posterior mantenimiento y reutilización de código. A veces es necesario realizar mejoras en los algoritmos o del aspecto de la interfaz y la dificultad de estas tareas depende en gran parte de la forma en que se hayan programado.

El modelo clásico de programación estructurada realiza un procesamiento lineal, esto es, un mismo módulo realiza todas las tareas. Si a esto, se le suma el hecho de que en la mayoría de las ocasiones, quien debe mantener o reutilizar el código no es quien lo escribió, se obtiene que la complejidad de las tareas de mantenimiento y reutilización del código escrito, es mayor que con otro paradigma. Por tanto, no es una opción válida para nuestro desarrollo.

En cambio, el patrón Modelo Vista Controlador facilita en gran parte las tareas de mantenimiento y reutilización dividiendo las aplicaciones en tres partes claramente diferenciadas, de manera que si se respeta estrictamente el patrón, no será necesario modificar todo su código cuando se pretenda mejorar o modificar la aplicación. La figura 3.1 representa esta división [23].

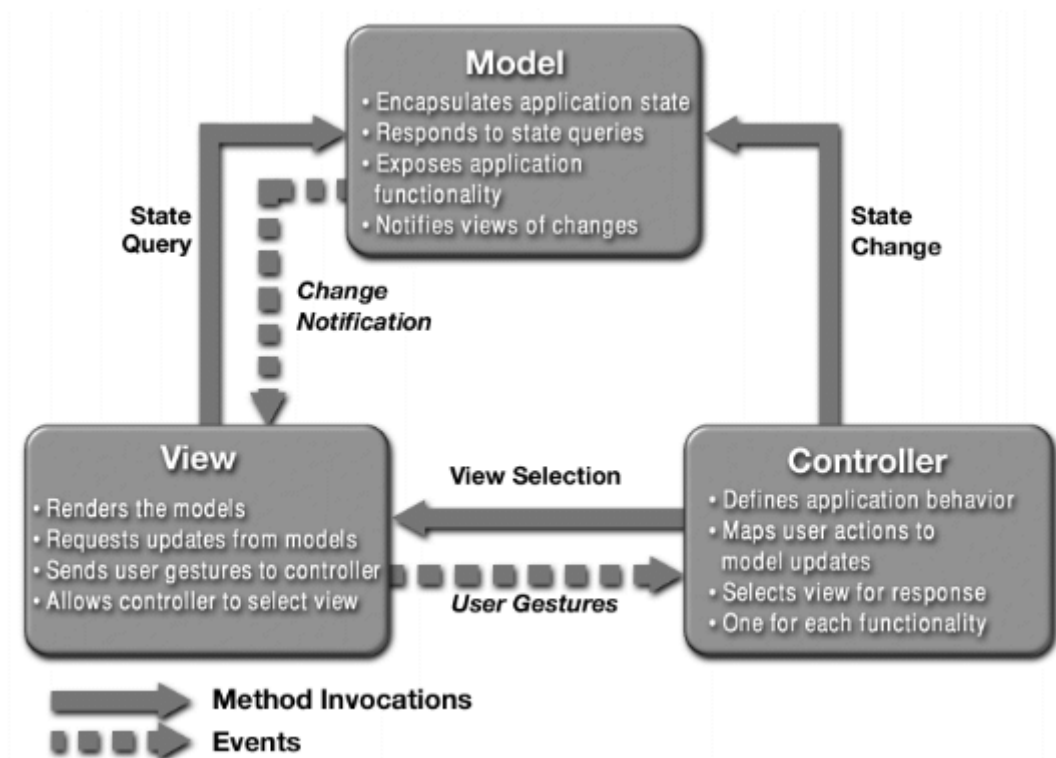


Figura 3.1. El patrón MVC.

Las tres capas en la que se estructura la aplicación bajo este patrón son:

- **Modelo.** El *Modelo* es la parte del patrón que se encarga de procesar los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- **Controlador.** El *Controlador* es la parte del patrón que proporciona significado a las órdenes del usuario, actuando sobre los datos manejados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.
- **Vista.** La *Vista* es la parte del patrón que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.

En este patrón, el procesamiento se reparte en estas tres capas. El controlador recibe un evento o solicitud, analiza esta petición y, basándose en su mapa de configuración, decide qué parte del modelo llevará a cabo la ejecución. El modelo realizará las operaciones oportunas de la lógica de negocio, accediendo a base de datos si es necesario. Una vez termina su ejecución, devuelve el flujo al controlador que se encargará de enviar el resultado a la vista.

El MVC es un patrón ampliamente utilizado en múltiples plataformas y lenguajes. Algunas de sus principales ventajas son:

- Las tres capas del modelo son independientes y pueden ser implementadas por separado.
- Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su lógica de negocio).
- Permite definir varias vistas para una misma aplicación, por ejemplo, una vista para web y otra para aplicación de escritorio.
- Mayor claridad de diseño.
- Facilita el mantenimiento. Cada una de las capas puede ser actualizada sin modificar el resto.

Sin embargo, este patrón también presenta algunos inconvenientes como son:

- Mayor complejidad. El patrón MVC introduce nuevos niveles de indirección y, por tanto, incrementa ligeramente la complejidad de la solución.
- Si el modelo cambia rápidamente, puede llegar a sobrecargar a las vistas con solicitudes de actualización o mantenerlas desactualizados.

La Figura 3.2 muestra el patrón MVC desarrollado con tecnologías Java.

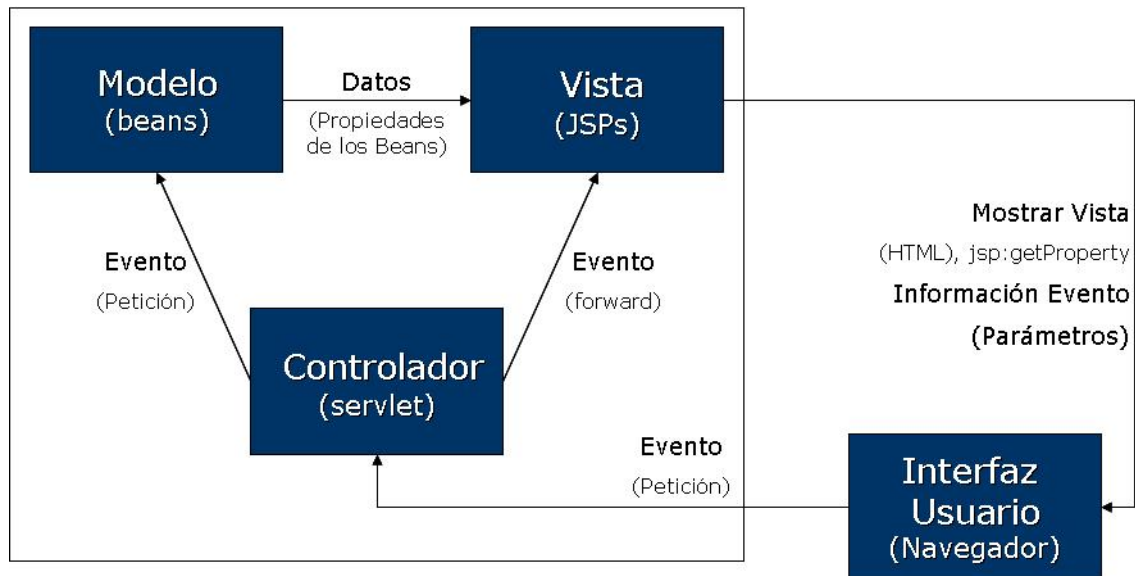


Figura 3.2. Patrón MVC - Tecnologías Java

### 3.2. Tecnologías utilizadas

#### 3.2.1 Struts

Struts es un proyecto nacido en mayo del año 2000 en el seno de la Apache Software Foundation [11]. Consiste en un entorno de desarrollo o Framework para desarrollar aplicaciones web que implementan el patrón Modelo Vista Controlador en el lenguaje Java.

Un Framework o marco de desarrollo es un término que se utiliza para definir a una serie de clases, interfaces, módulos, y librerías que cooperan conjuntamente para solucionar un determinado problema.

Struts [25], siguiendo el concepto de Framework, está formado por un conjunto de clases, servlets y una serie de librerías de etiquetas o taglibs que se utilizan en las páginas JSP, para acceder a objetos del modelo, de manera que estas páginas estén libres de código Java.

Struts ofrece soluciones para cada una de las partes del patrón MVC, así el modelo puede estar formado por cualquier clase Java y por clases que heredan del *Action* y *ActionForm* instanciadas desde clases propias de Struts, que son extensiones de un servlet. Los datos del modelo son pasados a la vista para ser mostrados al usuario. El papel del controlador lo realiza un servlet que se configura en un fichero XML, en este fichero se asignan las clases del modelo a peticiones del usuario y la vista que mostrará los datos. Struts también tiene mecanismos para validar los formularios de entrada, para el tratamiento de errores y para internacionalizar la vista.

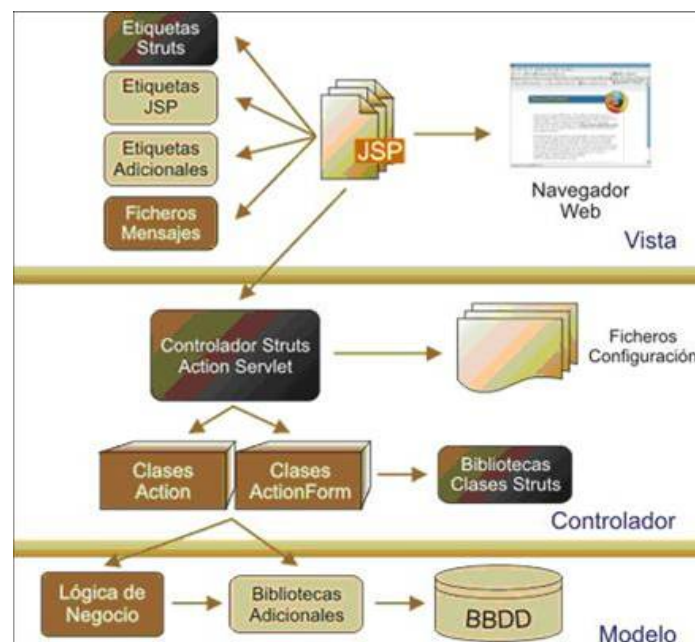


Figura 3.3. Implementación del patrón MVC con Struts [25]

En el patrón MVC (Figura 3.3) el servlet controlador está proporcionado por Struts, las páginas JSP son la vista, y la lógica de negocio de la aplicación reside en clases Java. El funcionamiento básico de Struts se detalla a continuación.

El servlet controlador de Struts dirige las solicitudes http de los navegadores web a otros objetos del Framework, especialmente subclases de *Action*. El controlador analiza el fichero de configuración, *struts-config.xml*, para realizar esta tarea. El objeto *Action* maneja la solicitud del cliente y devuelve un objeto *ActionForward*. Estos mapeos se encuentran, de nuevo, en el fichero de configuración de Struts y dirigen la vista hacia una página JSP. Por ejemplo, si la comprobación de un *login* tiene éxito, una acción podría querer mostrar el menú principal o un página de inicio, y en caso contrario, podría querer mostrar una página de error o de nuevo la página de validación de usuario.

En una aplicación Struts, la mayoría de la lógica de negocio se puede encapsular en JavaBeans. Una clase *Action* puede llamar a las propiedades y métodos de un JavaBean sin conocer su funcionamiento. Así la clase *Action* puede dedicarse al manejo de errores y dirección del flujo de ejecución y los beans son totalmente independientes del diseño de la vista. En aplicaciones sencillas, un objeto *Action* podría manejar la lógica de negocio asociada a una petición. Los JavaBeans también pueden usarse para manejar formularios de entrada.

Un problema clave en el diseño de aplicaciones Web es retener y validar los datos introducidos por el usuario, puesto que el protocolo HTTP es stateless, no mantiene información de estado. Con Struts, se puede definir un conjunto de clases que extienden de *org.apache.struts.action.ActionForm*, de manera que almacenen y validan de forma fácil los formularios de entrada. Los objetos *Action* reciben como parámetro estos objetos *ActionForm*. Los formularios Struts se declaran en el fichero de configuración de Struts, junto a los *ActionForward*, definidos para una solicitud. Cuando llega una solicitud, el controlador busca en el fichero *struts-config.xml* e instancia las clases *Action* y *ActionForm* que estén declaradas.

La Figura 3.4 muestra un diagrama con el flujo que ejemplifica la relación entre las clases de una aplicación Struts. La relación entre ellas es la siguiente; el usuario realizará una petición al servidor, por ejemplo, rellenar un formulario de datos y enviarlo al servidor. El controlador Struts buscará en su fichero de configuración que *Action* debe atender tal solicitud. La clase *Action* recibe como parámetro la información contenido en el formulario de entrada encapsulada en una clase que hereda de *ActionForm*. Por otro lado, al *Action* puede disponer de un *Bean* que encapsula todas las operaciones a realizar, y por tanto ella solo se encargará de la gestión de errores y el flujo de ejecución. Una vez acabadas las operaciones con los datos, la clase *Action* devolverá un objeto *ActionForward* que el controlador usará para decidir que página JSP mostrar a continuación.



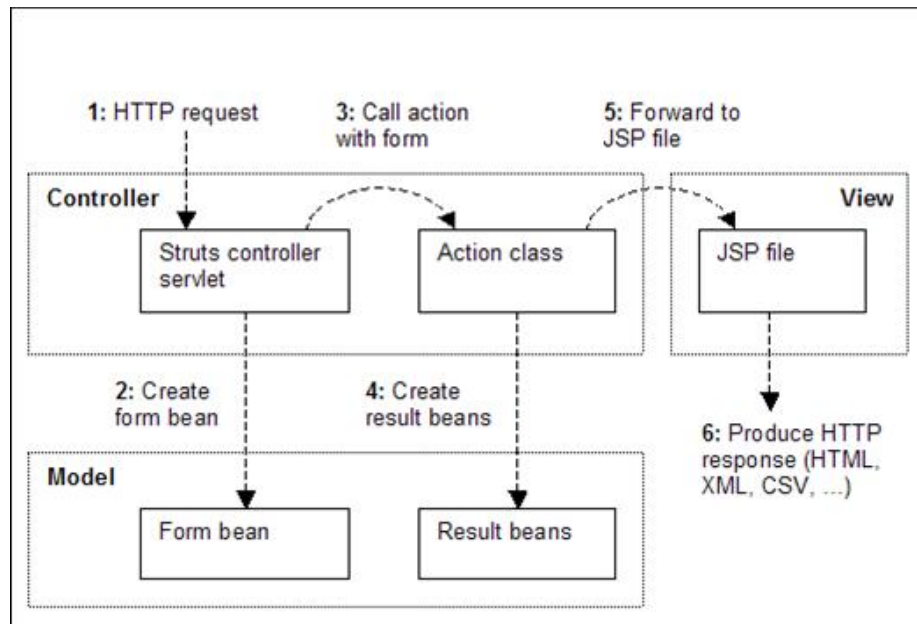


Figura 3.4. Flujo de una operación con Struts

### 3.2.2. JDOM

En la actualidad, la potencia del lenguaje XML es enorme y ha conseguido imponerse como un estándar de facto para el intercambio de datos entre sistemas. En cuanto al *e-learning*, varias especificaciones recomiendan su uso para la descripción de contenidos de aprendizaje. En concreto para este proyecto se han seguido las especificaciones de la IMS Global, todas ellas definen modelos de estructuras y comportamiento utilizando XML.

Existen varias librerías de código libre que permiten el tratamiento y procesamiento de ficheros XML de una forma fácil, intuitiva y potente. Para la implementación de esta herramienta, como se trabaja con Java, se ha utilizado el API JDOM [19].

JDOM es un API para leer, crear y manipular documentos XML de una manera sencilla y muy intuitiva para cualquier programador en Java, en contra de otras APIs tales como DOM (Document Object Model) y SAX (Simple API for XML), las cuales se idearon sin pensar en ningún lenguaje en concreto, de ahí que resulte un poco incómodo su utilización. DOM y SAX son dos especificaciones que como tal no podemos trabajar con ellas, pero sí con las implementaciones de dichas especificaciones, es decir, los parsers: Xerces [27], XML4j [28], Crimson [14], Oracle's parsers [24], etcétera.

La API JDOM no es un parser, de hecho, usa un parser para su trabajo, JDOM "solo" nos aporta una capa de abstracción en el tratado de documentos XML facilitándonos bastante la tarea de manera que no hay que ser expertos de DOM y SAX para poder trabajar con XML desde Java.

El API está formado por 5 paquetes. De entre ellas se van a comentar los más importantes para tener una visión general de JDOM:

- El paquete `org.jdom` destacamos las clases: **Document** que representará el documento XML, **Element** que representará el elemento o etiqueta que forma el documento, y la clase **Attribute** que representa los atributos que puedan tener los elementos.
- El paquete `org.jdom.input` que albergará las clases **Builder** para construir los documentos XML.
- El paquete `org.jdom.output` que albergará las clases que utilizaremos para dar salida a nuestra clase **Document**.

Para trabajar con JDOM hay que indicarle que parser vas a utilizar, en el caso de este proyecto se utiliza Xerces.

### 3.2.3 XSLT

Al igual que XML, XSLT [29] es un lenguaje de programación. Forma parte de la *trilogía transformadora* de XML, compuesta por las CSS (*Cascading Style Sheets*, hojas de estilo en cascada), que permite dar una apariencia en el navegador determinada a cada una de las etiquetas XML; XSLT (*XML Stylesheets Language for Transformation*, o lenguaje de transformación basado en hojas de estilo); y XSL:FO, (Formatting Objects, objetos de formato de datos), o transformaciones para fotocomposición, o, en general, para cualquier cosa que no sea XML, como por ejemplo HTML o PDF. Por tanto se ha utilizado este lenguaje para la visualización de los contenidos que siguen la especificación IMS QTI de la IMS Global (especificación descrita en XML).

XSLT es pues, un lenguaje que se usa para convertir documentos XML en otros documentos XML; puede convertir un documento XML que obedezca a un DTD a otro que obedezca otro diferente, un documento XML bien formado a otro que siga un DTD, o, lo más habitual, convertirlo a "formatos finales", tales como WML (usado en los móviles WAP) o XHTML.

Los programas XSLT están escritos en XML, y generalmente, se necesita un procesador de hojas de estilo, o *stylesheet processor* para procesarlas, aplicándolas a un fichero XML.

El estilo de programación con las hojas XSLT es totalmente diferente a los otros lenguajes a los que estamos acostumbrados (tales como Java o C++), pareciéndose más a "lenguajes" tales como el AWK, o a otros lenguajes funcionales, tales como ML o Scheme. En la práctica, eso significa dos cosas:

- *No hay efectos secundarios.* Una instrucción debe de hacer lo mismo cualquiera que sea el camino de ejecución que llegue hasta ella. Es decir, no va a haber variables globales, ni bucles en los que se incremente el valor de una variable, o tenga un test de fin de bucle, ni nada por el estilo. Aunque permite la posibilidad de emular recursión.
- *La programación está basada en reglas:* cuando ocurre algo en la entrada, se hace algo en la salida. Esto en XSLT se conoce como *Templates*.

XSLT, es la única alternativa cuando uno quiere adaptar un contenido descrito con XML a diferentes clientes (por ejemplo, móviles de diferente tamaño, diferentes navegadores), y la mejor alternativa cuando uno quiere procesar documentos XML (aunque haya otras: filtros SAX, expresiones regulares...). Otra alternativa, sobre todo si se está trabajando ya con un documento XML en forma de DOM (Document object model) es trabajar directamente sobre él. En este caso, de todas formas, se pueden usar transformaciones XSL, sólo que se aplicarán en memoria, en vez de leerlas desde un fichero.

Lo que consiguen las hojas de estilo es separar la información (almacenada en un documento XML) de su presentación, usando en cada caso las transformaciones que sean necesarias para que el contenido aparezca de la forma más adecuada en el cliente. Es más, se pueden usar diferentes hojas de estilo, o incluso la misma, para presentar la información de diferentes maneras dependiendo de los deseos o de las condiciones del usuario.

Aparte del hecho habitual de procesar documentos XML, XSLT es un lenguaje de programación, y por tanto se podría hacer cualquier cosa con ellas; es decir, se podría implementar cualquier algoritmo. Pero en este caso me interesa más como simple herramienta de transformación de XML.

### 3.3 Especificaciones seguidas

#### 3.3.1 IMS Question & Test Interoperability

Los contenidos educativos manejados en el proyecto siguen la especificación IMS QTI 1.2 [18], ya que la aplicación "MD2: Sistema de búsqueda y composición de contenidos", creada como resultado de este trabajo se centra en los contenidos basados en exámenes conformes con dicha especificación.

Esta especificación contempla una estructura básica que describe la forma de representar preguntas individuales o ítems (*assessment item*) y gestionar evaluaciones o exámenes completos (*assessment*). Su objetivo es conseguir que tanto las evaluaciones como los resultados sean intercambiables entre los

diferentes LMS. Así, podemos disponer de almacenes de preguntas y bases de datos con los resultados obtenidos por los alumnos a los que cualquier sistema de enseñanza electrónica podrá acceder.

Con este propósito se plantea y se documenta un formato de contenido para almacenar las preguntas o ítems independientemente del sistema o herramienta de autoría utilizada para crearlas. Esto permite, por ejemplo, el uso de las mismas preguntas en diversos LMS o en sistemas de evaluación electrónica, o la integración en un único LMS de preguntas o exámenes desarrollados con distintas herramientas. Por otro lado se propone un sistema coherente para que los sistemas puedan informar de cuál es el resultado de una evaluación.

En la Figura 3.5 se muestra una pregunta con respuesta múltiple, en este caso con una sola respuesta correcta.

The image shows a digital interface for a multiple-choice question. At the top, the title 'UNATTENDED LUGGAGE' is displayed in a bold, serif font. Below the title, the instruction 'Look at the text in the picture.' is shown. In the center, there is a rectangular box containing the text 'NEVER LEAVE LUGGAGE UNATTENDED'. Below this box, the question 'What does it say?' is presented. Underneath the question, there are three possible answers, each in a separate row with a radio button to its right:

What does it say?	
You must stay with your luggage at all times.	<input type="radio"/>
Do not let someone else look after your luggage.	<input type="radio"/>
Remember your luggage when you leave.	<input type="radio"/>

Figura 3.5. Ejemplo de pregunta de un examen tomado del sitio de la especificación IMS QTI [18]

Dentro de la especificación un ítem incluye la pregunta que se presenta al usuario y puede incluir otra información necesaria para el procesamiento de la respuesta o puntuación, retroalimentación instantánea o consejos para su realización, y otros mecanismos para mejorar el examen y/o la evaluación (Figura 3.6). QTI trata de ser pedagógicamente neutral y proporciona un gran conjunto de preguntas que habitualmente se utilizan en las evaluaciones,

tales como elección verdadero/falso, elección múltiple con respuesta única, elección múltiple con varias respuestas válidas, rellenar campos en blanco, ordenar objetos, relacionar objetos, etc. Además permite definir nuevos tipos de preguntas si fuera necesario.

Las preguntas se agrupan en secciones, que a su vez se agrupan para formar una evaluación o examen. Una evaluación, examen o test es una colección de secciones que agrupan items y que además contiene información sobre cómo secuenciar los ítems (presentación secuencial o se “barajan” las preguntas antes de presentarlas) y como combinar sus evaluaciones individuales para obtener la evaluación final. Esto permite, por ejemplo, definir cuál es el número de preguntas que se deben responder correctamente para que el examen se considere aprobado.

En la Figura 3.6 se muestra el modelo de objetos de la especificación IMS QTI 1.2 y la relación entre dichos objetos.

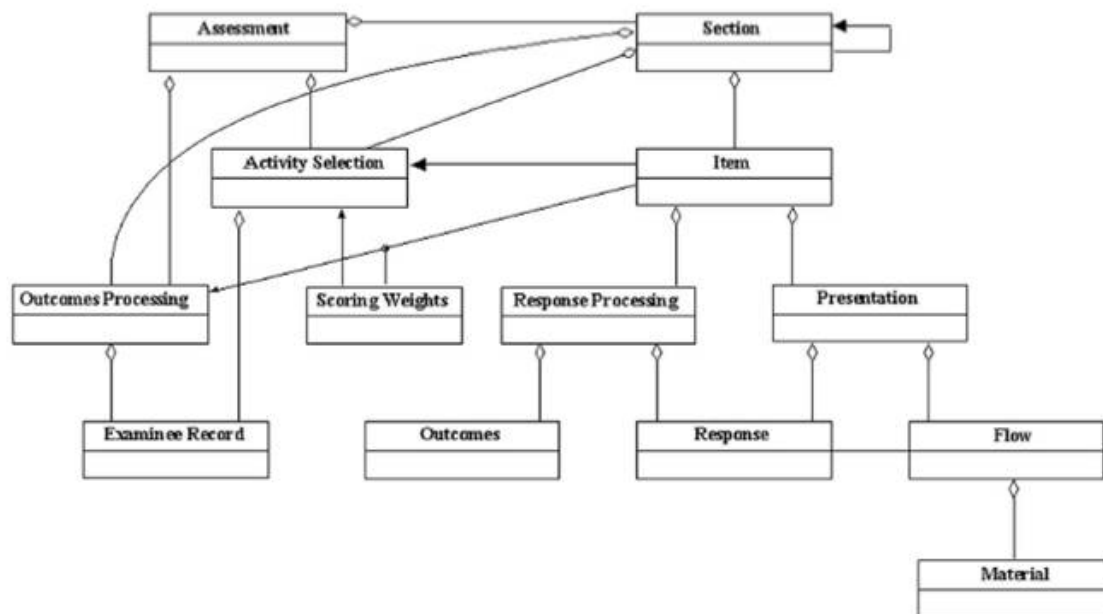


Figura 3.6. Modelo de objetos de IMS QTI [18]

Los objetos en este modelo y sus comportamientos clave son los siguientes:

- *Assessment*. El objeto que representa la estructura de datos de la evaluación.
- *Section*. El objeto que representa una parte o sección de la estructura de datos de la evaluación.
- *Item*. El objeto que representa la estructura de datos del contenido.
- *Activity Selection*. Selección de la próxima actividad determinada por los avances y resultados obtenidos hasta el momento.

- *Outcomes Processing*. La recopilación de todos los productos de evaluación para producir una evaluación global.
- *Scoring Weights*. Los pesos que se asignan a los resultados de salida de la respuesta.
- *Response Processing*. La tramitación y evaluación de las respuestas del usuario.
- *Presentation*. La presentación del contenido y las posibles respuestas.
- *Examinee Record*. El conjunto de resultados cotejados durante todo el proceso. Se trata del histórico sobre el avance del estudiante.
- *Outcomes*. El conjunto de resultados que han de ser evaluados por el *Response Processing*. Estos determinan la puntuación que se aplicará a la respuesta en las evaluaciones.
- *Response*. Las respuestas que son aportadas por el usuario de los contenidos: el estudiante que realiza el test.
- *Flow*. Es la estructura subyacente que define el bloque de relación entre los diferentes materiales.
- *Material*. El contenido que se visualiza.

La Figura 3.7 muestra la relación entre los tres principales objetos de datos, que son, *item*, *section* y *assessment*.

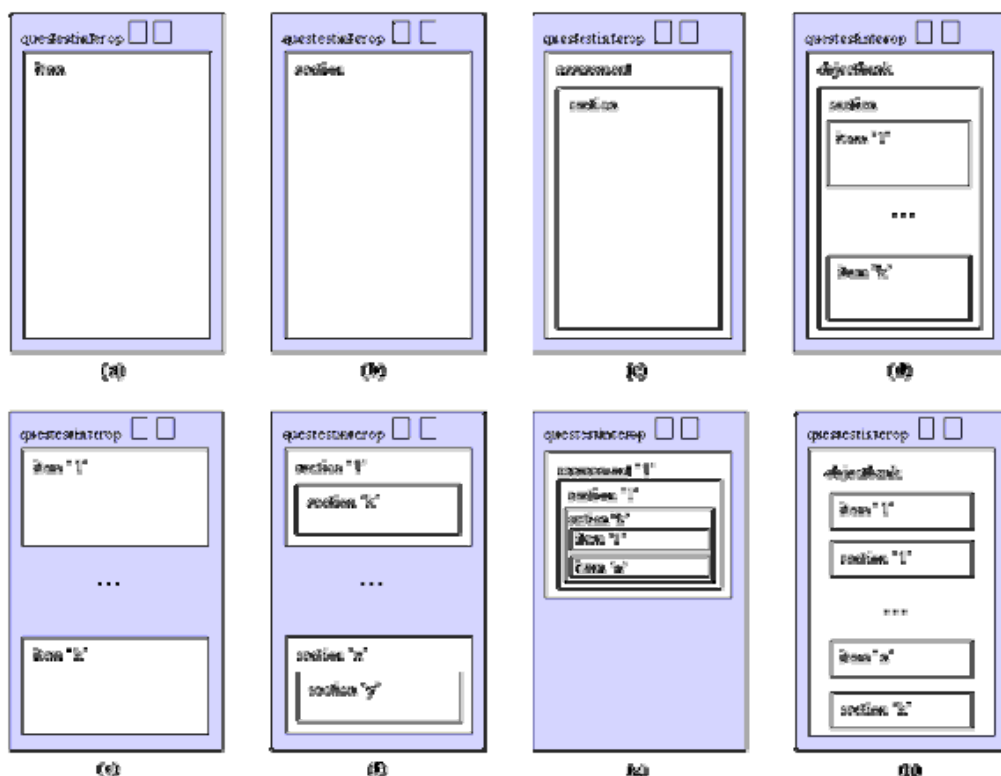


Figura 3.7. Relación entre los principales objetos de datos de IMS QTI 1.2 [18]

A continuación se trata de explicar los elementos más importantes de esta especificación, en especial aquellos que influyen en la visualización del contenido o examen.

Como se ve en la Figura 3.8 el objeto raíz de la especificación es *questestinterop* que simplemente es la raíz del documento XML y que contiene los objetos más importantes.

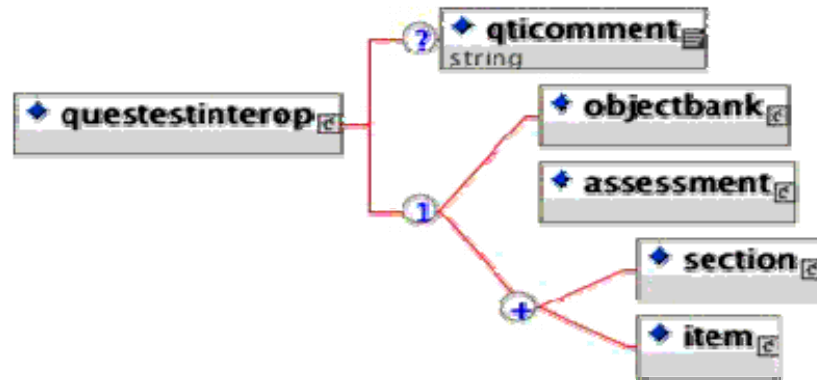
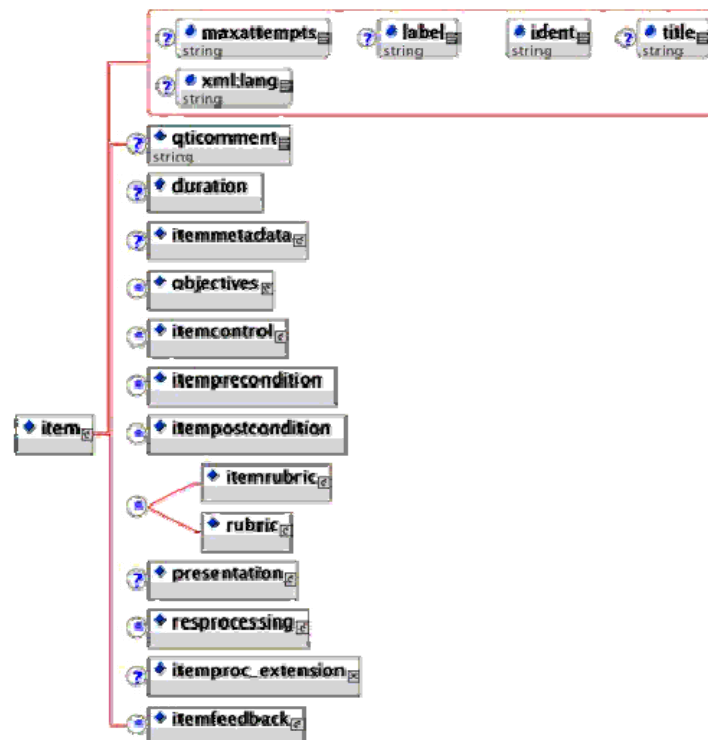


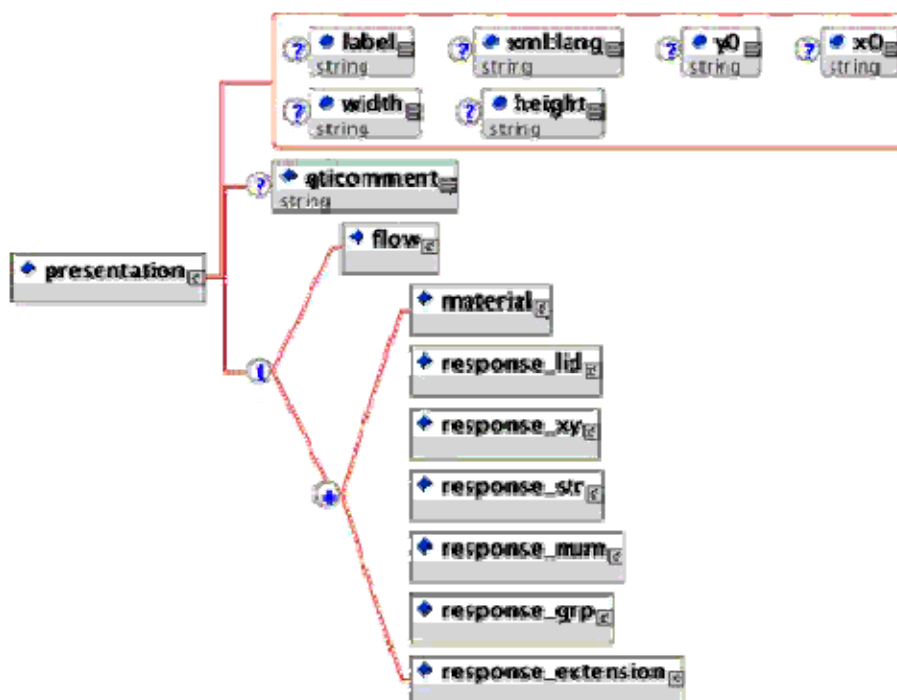
Figura 3.8. Raíz del documento XML de la especificación IMS QTI [18]

Esta raíz contiene los elementos más importantes del contenido de los que destacaremos:

- *Assessment*. Es el objeto que representa el examen completo y contiene los objetos necesarios para la visualización del contenido.
- *Section*. Es una agrupación de ítems que contiene información que los caracteriza y que incluye la relación entre los ítems y los criterios de selección de esos ítems.
- *Item*. Un *item* contiene toda la información para la presentación de una pregunta al participante y su posterior transformación en la respuesta al usuario. La estructura del *item* incluye la cuestión real y su formato de presentación, la gama de posibles respuestas, las formas en que las respuestas se van a procesar, las posibles soluciones y sugerencias para el tema y comentarios. La Figura 3.9 muestra el esquema que sigue este objeto y los objetos que contiene.

Figura 3.9. Esquema XML del objeto *item* [18]

El objeto *item* contiene el objeto *presentation* que nos va a indicar qué tipo de pregunta es y cómo se visualiza. También contiene la respuesta correcta que se podrá procesar en la parte de *resprocessing*. En la Figura 3.10 se muestra el esquema en forma de árbol de los objetos que componen el objeto *presentation*.

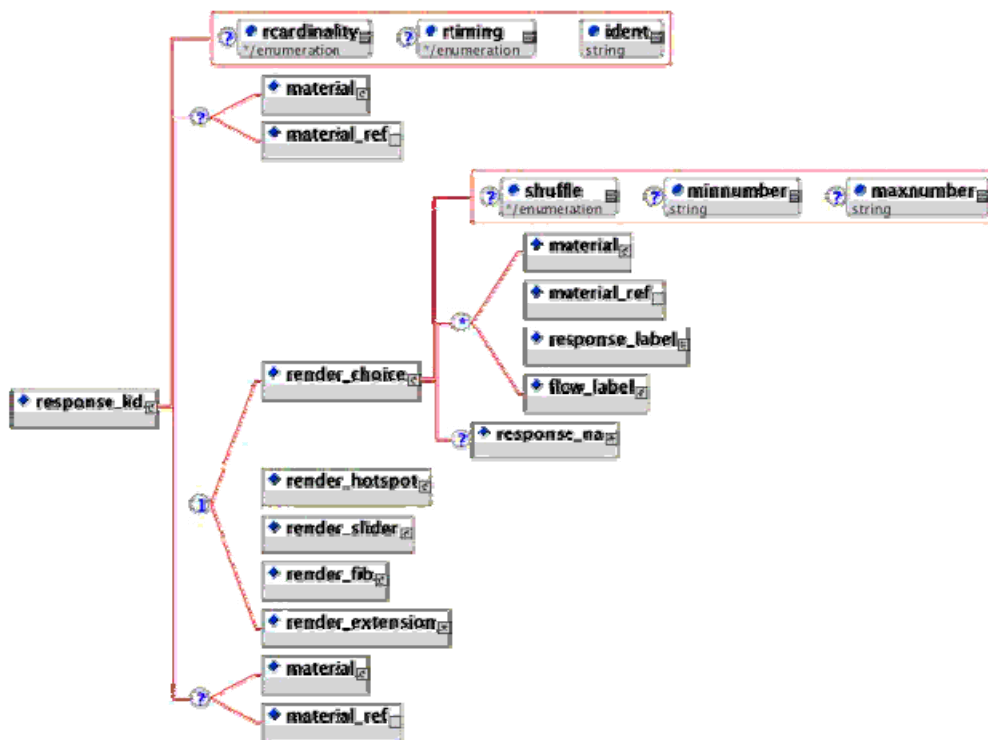
Figura 3.10. Esquema XML del objeto *presentation* [18]



Según la especificación dentro de la presentación del contenido se especifican seis objetos distintos que se van a corresponder con el tipo de pregunta, más concretamente con la identificación de la respuesta. Estos objetos son los siguientes:

- *Response\_lid*. Son aquellas preguntas en las que la respuesta va a venir determinada por un identificador lógico (Logical Identifier).
- *Response\_xy*. Son aquellas preguntas en las que la respuesta va a venir determinada por las coordenadas x e y dentro de una imagen.
- *Response\_str*. En este tipo de preguntas la respuesta no se determinará de ninguna manera sino que será directamente la cadena de texto que cumpla con la respuesta correcta.
- *Response\_num*. De manera análoga al tipo *response\_str*, la respuesta a este tipo de preguntas será directamente un número entero o decimal, que no poseerá ningún tipo de identificador.
- *Response\_gpr*. En este tipo, la respuesta viene formada por un grupo lógico de elementos que poseen una relación entre ellos.
- *Response\_extension*. Por último, las respuestas de este tipo se corresponden a extensiones de la especificación que entidades privadas hayan hecho, de manera que aplicaciones propietarias podrán incluir identificaciones de respuestas de propio creación.

Cada tipo de respuesta podrá ser visualizado de distintas maneras, se distinguen hasta cinco tipos de visualización, pero no todos los tipos de respuesta pueden visualizarse de todas las maneras. La figura 3.11 muestra el esquema XML de un tipo de respuesta, en este caso *response\_lid*, como ejemplo de las posibles formas de visualización.

Figura 3.11. Esquema XML del objeto *response\_id* [18]

Los tipos de visualización que contempla la especificación son los siguientes:

- *Render\_choice*. Se presenta la pregunta con varias opciones para la posible respuesta y puede ser multi-respuesta o el clásico verdadero/falso.
- *Render\_hotspot*. La pregunta se basa en una imagen y la respuesta estará contenida en alguna coordenada dentro de la propia imagen.
- *Render\_fib*. La visualización de este tipo consiste en dejar huecos en blanco para que el usuario escriba la posible respuesta que puede ser numérica o cadenas de texto.
- *Render\_slider*. En este tipo de presentación la pregunta se basa en una barra que es un elemento gráfico que permite seleccionar un valor moviendo un indicador o, en algunos casos, el usuario puede hacer clic sobre algún punto del *slider* para cambiar hacia ese valor.
- *Render\_extension*. Por último, la especificación da la posibilidad a posibles visualizaciones implementadas en aplicaciones propietarias.

En la Figura 3.12 se puede observar un ejemplo de una pregunta basada en la presentación *render\_slider*.

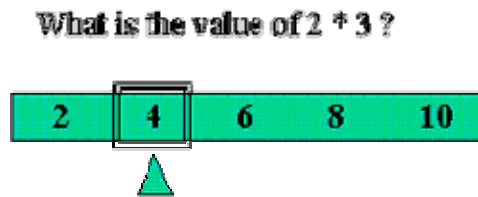


Figura 3.12. Ejemplo de pregunta con visualización *render\_slider* [18]

### 3.3.2 IMS Content Packaging

La otra especificación que está soportada por la aplicación que se ha diseñado es el IMS Content Packaging [15]. El objetivo de esta especificación es permitir la distribución de contenidos reutilizables e intercambiables, es decir, describe el modo en el que se debe empaquetar el contenido educativo para que pueda ser procesado por otro sistema LMS diferente. Ofrece una forma de empaquetar (en un archivo comprimido tipo .zip) los contenidos educativos tales como cursos individuales, conjuntos de cursos, o cualquier tipo de recurso necesario en el proceso educativo (por ejemplo, evaluaciones o exámenes). Al distribuir una serie de contenidos empaquetados según el Content Packaging de IMS, existe un documento fundamental que es el *Manifiesto*. Dicho documento es un fichero XML en el que se describe la estructura de los contenidos incluidos en el paquete (Figura 3.13). Dicha descripción se realiza a dos niveles diferentes:

- Por un lado, se describe cada uno de los *Recursos* del paquete. En una primera aproximación se puede hacer una relación casi directa entre un Recurso y un fichero con contenidos visualizables (Ej. un Objeto de Aprendizaje) como pueden ser un conjunto de ficheros HTML, animaciones en Flash, imágenes, etc. En realidad, en cada Recurso se puede incluir información sobre los ficheros que componen dicho Recurso, el tipo de los mismos (que puede ser uno de los tipos ya definidos por el estándar o una extensión de los propuestos) y, opcionalmente, metadatos con información adicional sobre dicho Recurso.
- Por otro lado, en el Manifiesto se describe como están organizados dichos Recursos, es decir, como se estructuran los contenidos del paquete. Esto se implementa mediante las *Organizaciones*. Una organización es una vista (o recorrido) de una posible ordenación jerárquica (actualmente en forma de árbol) de los *Recursos* de un paquete. El estándar permite que un Manifiesto contenga distintas organizaciones sobre los *Recursos* del paquete, dando así lugar a distintas vistas o “cursos” a partir de los mismos contenidos. El elemento básico de estructuración que se usa al definir las organizaciones son los Ítems. A cada Ítem se le puede asociar un

Recurso, de modo que el árbol de Ítems es, efectivamente, una estructuración de los Recursos del paquete.

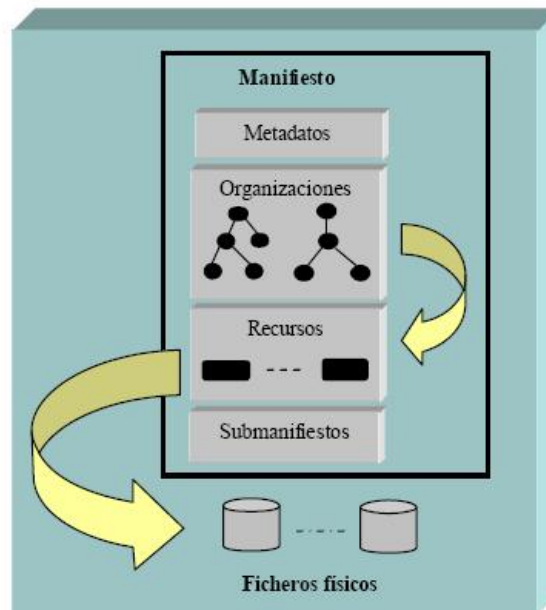


Figura 3.13. Esquema de un manifiesto [15]

En resumen, el Manifiesto es un fichero XML que describe y organiza los contenidos de un paquete, añadiendo información adicional en forma de metadatos que pueden ser procesados y aprovechados en tareas de catalogación de contenidos (Figura 3.14).

En la Figura 3.14 se muestra un ejemplo de un Manifiesto XML que sigue el estándar IMS Content Packaging.

```

<?xml version="1.0"?>
<manifest identifier="MANIFEST1"
xmlns="http://www.imsproject.org/xsd/ims_cp_rootv1p1">
<metadata>

</metadata>
<organizations default="TOC1">
<organization identifier="TOC1">
<title>Big Title</title>
<item identifier="ITEM1" identifierref="RESOURCE1">
<title>Lesson 1</title>
<item identifier="ITEM2" identifierref="RESOURCE2">
<title>Introduction 1</title>
</item>
<item identifier="ITEM3" identifierref="RESOURCE3">
<title>Content 1</title>
</item>
<item identifier="ITEM4" identifierref="RESOURCE4">
<title>Summary 1</title>
</item>
</item>
</organization>
</organizations>
<resources xml:base="http://repositoy.imsproject.org/foo/bar/">
<resource identifier="RESOURCE1" type="webcontent "
href="lesson1.htm"/>
<resource identifier="RESOURCE2" type="webcontent "
href="introl.htm"/>
<resource identifier="RESOURCE3" type="webcontent "
href="content1.htm"/>
<resource identifier="RESOURCE4" type="webcontent "
href="summary1.htm"/>
</resources>
</manifest>

```

Figura 3.14. Ejemplo de manifiesto XML [15]

Finalmente, para la distribución e intercambio efectivo de los cursos, la especificación indica que debe crearse un *Archivo de Intercambio de Paquetes* (*Package Interchange File*, o simplemente PIF). El "PIF" es un archivo que alberga en su interior el manifiesto y los recursos que se referencian en dicho manifiesto. Por tanto, podemos decir que es un paquete comprimido y con un formato de intercambio en formato .zip. La funcionalidad de exportación a PIF o de importación de un PIF se encuentra en muchos de los LMS tanto comerciales (Ej. WebCT) como de software libre (Ej. Moodle, .LRN, Dokeos, Claroline).

### 3.3.3 IMS Learning Resources Meta-data

Los metadatos proporcionan descripciones, propiedades e información sobre los objetos de aprendizaje (OA) que permiten caracterizarlos, de forma que se simplifica su uso y gestión. De forma coloquial, lo que se busca mediante

esta información complementaria es poder saber cuál es el contenido y el propósito de un OA sin tener que acceder a dicho contenido. Por tanto, los metadatos aportan información orientada a hacer más eficiente la búsqueda y utilización de los recursos. Los metadatos se pueden aplicar tanto a OA concretos como a cursos completos o a partes del curso (como se puede ver en el esquema del manifiesto de la (Figura 3.13).

Actualmente LOM (*IEEE Learning Object Meta-Data*) es el estándar de *e-learning* formalmente aprobado que goza de mayor aceptación (estándar IEEE 1484.12.1 - 2002), y que ha sido adoptado en la especificación de *IMS Learning Resource Metadata* [17]. De hecho LOM se basa en los esfuerzos previos hechos para la descripción de recursos educativos en los proyectos ARIADNE, IMS y Dublin Core.

El objetivo de LOM es la creación de descripciones estructuradas de recursos educativos. Su modelo de datos especifica qué aspectos de un objeto de aprendizaje deberían ser descritos y qué vocabularios se pueden utilizar en dicha descripción.

Esta es una descripción jerárquica con nueve apartados principales que agrupan el resto de campos. A continuación describimos cada una de estas categorías:

- *General*. Aquí se describe el objeto educativo. Incluye campos como identificador del OA, título, descripción, etc.
- *Lifecycle*. Almacena un histórico del objeto y su estado actual. Detalla quiénes han interactuado con este objeto desde que fue creado, y el tipo de interacción que han realizado.
- *Meta- Metadata*. Agrupa información sobre los metadatos. Esto puede parecer redundante a primera vista pero resulta muy interesante tener información como quién ha contribuido a la creación de los metadatos y el tipo de contribución que ha realizado.
- *Technical*. Incluye la información técnica del recurso de aprendizaje, tal como tamaño, ubicación, o formato en el que se encuentra. Además, en este elemento se almacenan los posibles requisitos técnicos necesarios para poder usar el objeto al que se refieren los metadatos.
- *Educational*. En este elemento se encuentran las diferentes características pedagógicas del objeto. Típicamente se incluyen campos como tipo de recurso - ejercicio, diagrama, figura -, nivel de interactividad entre el usuario y el objeto -alta, media, baja-, o el contexto de uso del recurso - universidad, enseñanza primaria, doctorado-, entre otros.
- *Rights*. Se incluyen los detalles sobre la propiedad intelectual del recurso. También se detallan las condiciones de utilización y el precio en caso de tenerlo.

- *Relation*. Explica el tipo de relación que tiene el recurso de aprendizaje con otros OA. Posee un par nombre-valor en el que detalla el nombre del OA relacionado y el tipo de relación -es parte de, está basado en, etc. -.
- *Annotation*. Incluye comentarios sobre la utilización del OA, además de su autor y la fecha de creación.
- *Cassification*. Nos informa si el OA pertenece a algún tema en concreto. Por ejemplo, es aquí dónde se almacenaría que un OA se refiere a Física o a Historia. Permite tanto detalle cómo se quiera mediante anidamiento de temas.

El modelo de datos especifica también qué elementos de la descripción pueden repetirse (Ej. Classification). Además, hay unos campos en los que el tipo de contenido es libre, es decir se puede poner cualquier cadena de texto (para la cuál se puede especificar además el idioma) y hay otros campos en los que se dispone de un conjunto de valores concretos entre los que se puede elegir (es decir, se tiene un vocabulario controlado).

Como hemos mencionado antes existe otra especificación anterior cuyo nivel de aceptación es también muy amplio: el Dublin Core. Nacida con el objetivo de describir recursos de carácter genérico en la Web, también ha sido adoptado por la comunidad educativa con el fin de adjuntar información complementaria a los recursos educativos.

En este caso, y frente a los más de 70 campos de LOM, los 15 metadatos básicos de Dublín Core para un recurso educativo son: título, autor, tema o palabras clave, descripción, editor, otros colaboradores, fecha, tipo de recurso, formato, identificador, fuente, idioma, relación con otros recursos, cobertura, y derechos. En el documento del propio estándar LOM se incluye un apéndice comparando ambas especificaciones.

### 3.3.4 MD2

El análisis de las soluciones actuales a la problemática del desarrollo de los materiales didácticos llevó a la autora de la solución MD2 [6] a plantear que: El desarrollo puede ser soportado de forma efectiva si se cuentan con los medios que permitan describir sus componentes fundamentales, es decir, los contenidos y la estrategia pedagógica, así como sus propiedades deseables, tales como el carácter reutilizable, la calidad, la usabilidad, la utilidad o valor pedagógico y el cumplimiento con los estándares y especificaciones de *e-learning*. Además es preciso contar con un método que guíe las diferentes etapas del desarrollo: selección, composición y evaluación. Por lo tanto, la hipótesis de trabajo, como solución a esta problemática, fue detallada por la autora de la siguiente forma:

1. Si tenemos un modelo que describa de manera general y formal los requisitos de los contenidos del material, tanto de carácter pedagógico, como técnico y de calidad, entonces estos descriptores se podrían utilizar como criterios para la selección de los contenidos más adecuados así como para controlar criterios de calidad como completitud, valor pedagógico, coherencia, exactitud o adecuación de tales contenidos.
2. Si el modelo también describe las estrategias pedagógicas utilizando los estándares y especificaciones de *e-learning*, entonces se contará con información acerca de la estructura para la implementación del material y sobre cómo se pueden conectar o asociar los contenidos a dicha estructura.
3. Por tanto un método para el desarrollo de materiales didácticos, capaz de procesar la información facilitada por el modelo, puede resolver de forma sistemática las interrogantes que surgen en cada una de las diferentes etapas del desarrollo. Método y modelo proporcionarán las bases para poder crear una herramienta de autoría de carácter generativo.

La idea es que los usuarios de dicha herramienta puedan especificar los requisitos del material con un alto grado de abstracción sin necesidad de conocer los detalles de los estándares y su implementación, y la herramienta, utilizando la información del modelo, guiada por el método sea capaz de inferir los detalles de más bajo nivel del diseño del material, permitiendo ensamblar los componentes del material en tiempo de ejecución o entrega y la generación de las debidas anotaciones semánticas, que incluyen elementos del modelo como descriptores del material y la razones de su desarrollo, favoreciendo su localización y futura reutilización. A través de esta herramienta también se comprobará la calidad y usabilidad del material obtenido a partir de la valoración de la completitud, coherencia, exactitud y valor pedagógico de sus contenidos así como su capacidad para “soportar-apoyar” eficazmente la consecución exitosa de los objetivos de aprendizaje de un proceso educativo.

Con vistas a comprobar la validez de la hipótesis planteada se ha diseñado una solución que abarca los objetivos anteriores. La solución lleva el nombre de MD2, acrónimo de Método de Desarrollo de Materiales Didácticos y su automatización. Esta solución se ha desarrollado dentro del marco de dos proyectos subvencionados por el Plan Nacional I+D del Ministerio de Ciencia y Tecnología: MD2: Método de desarrollo de materiales educativos extendidos y su automatización (TIC2003-03654) y MODUWEB: Desarrollo basado en modelos de sistemas web usables de Tele-Educación (TIN2006-09678).

Esta solución se basa en un marco de desarrollo compuesto por un modelo, un método de desarrollo basado en el modelo y la arquitectura de una herramienta que implementa el método. El modelo MD2, mostrado en la Figura 3.15, describe de forma general los requisitos de contenido (relativos a



un dominio de conocimiento), pedagógicos, de soporte técnico y de calidad de los materiales didácticos a través de descriptores o elementos del modelo agrupados en cuatro vistas: Dominio de Conocimiento (DC), Pedagógica (P), Soporte (S) y Calidad-Usabilidad (C-U).

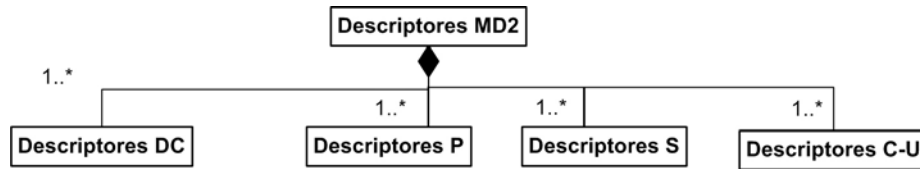


Figura 3.15. Vista del Modelo MD2

La idea es que los descriptores del modelo y las relaciones existentes entre ellos nos permitan definir correspondencias entre las descripciones de alto nivel técnico provenientes de los estándares *e-learning* con descripciones de los requisitos del material, más simples y cercanos al lenguaje de sus creadores. Las relaciones y correspondencias entre algunos elementos de las vistas DC, P y S facilitan la información necesaria para guiar el desarrollo en sus fases de selección y composición definiendo cuales componentes deben elegirse para el material y cuál es la estructura de presentación y entrega más apropiada según el estándar elegido. Por otra parte, un conjunto de los elementos de la vista C-U se utilizarán para controlar y valorar la usabilidad, el valor pedagógico y la calidad del material creado durante la etapa de evaluación. Los EMLs (Educational Modeling Languages) son soluciones que permiten describir y definir contenidos y procesos educativos desde una perspectiva pedagógica, que se han denominado en este trabajo componentes del material didáctico. A diferencia de los EMLs, el meta-modelo MD2 tiene como objetivo facilitar el desarrollo de materiales, cuya estructura de presentación y entrega se puede describir utilizando especificaciones o estándares *e-learning* en especial las especificaciones IMS LD y IMS QTI.

El método MD2 se ha diseñado para dar soporte a las 3 etapas del desarrollo de materiales: selección de recursos, composición del material y evaluación. Está formado por 5 pasos: Entrada de requisitos, Selección, Composición, Evaluación y Generación de anotaciones semánticas. La entrada de requisitos o datos del material, utiliza un conjunto de 12 pasos que se basan en simples preguntas a los desarrolladores sobre los requisitos del material. Estos pasos siguen un orden lógico, en primer lugar se cuestiona sobre la vista DC (paso 1), esto facilitará información pedagógica sobre el dominio de conocimiento, que será la base sobre la que se desarrollará el proceso de enseñanza y aprendizaje. En segundo orden estarán las preguntas relativas a los elementos de la vista P (pasos del 2 al 5) que determinarán los requisitos específicos del material de carácter pedagógico. A continuación se presentan las preguntas relacionadas con la vista S que definirán los requisitos del soporte tecnológico del material (pasos del 6 al 12). Las respuestas a cada una de estas preguntas serán los datos de entrada a los mecanismos de inferencia basados en las restricciones y axiomas del modelo MD2 que ayudarán a localizar y en algunos casos a crear, los recursos de contenidos durante el paso correspondiente a la

etapa de selección. Los mecanismos de inferencia también utilizan un subconjunto de estas respuestas para obtener las correspondencias entre los requisitos establecidos y los elementos del modelo de información del estándar de entrega y publicación, de forma que se define y pueda completar la estructura de presentación del material durante el paso correspondiente a la etapa de composición del desarrollo. El paso de evaluación del método se dedica a evaluar la completitud, coherencia, riqueza y estructura de los contenidos del material y la capacidad del mismo como apoyo efectivo del proceso educativo determinada por la valoración de la usabilidad del material. Los resultados de estas evaluaciones serán de gran utilidad para los desarrolladores puesto que les proporcionarán cierto grado de confianza sobre la calidad y efectividad del material recién creado. Por último, durante el paso 5, generación de anotaciones semánticas, todos los valores asignados a los elementos del modelo MD2 se almacenarán de forma automática como anotaciones semánticas del material al concluir la etapa evaluación del desarrollo. De acuerdo con las definiciones del meta-modelo MD2 para las correspondencias entre los valores de los elementos de las vistas DC, P y S con elementos de 4 categorías del estándar de anotación IEEE LOM (IMS LRMI): *General, Educational, Technical y Classification*. De esta forma se asegurará la futura localización y recuperación del material favoreciendo su posible reutilización.

El presente proyecto tiene el objetivo de implementar la solución MD2 para el caso en que los materiales didácticos son actividades de evaluación basadas en la especificación IMS QTI.

---

## 4. SOLUCIÓN DETALLADA

---

En este capítulo se describe el análisis y el diseño de la solución realizada.

### *4.1 Análisis*

#### 4.1.1 Análisis de Requisitos

Una de las primeras fases realizadas en el proyecto, tal como se describe en la sección 1.2, fue la recogida y análisis de requisitos de usuario. En esta fase se mantuvieron reuniones periódicas con la tutora del proyecto, con el objetivo de definir las necesidades que el sistema a desarrollar debía cubrir.

A continuación se presentarán los requisitos asociados al proyecto y se realizarán comentarios sobre su grado de cumplimiento en el sistema desarrollado objeto del presente estudio. Los requisitos se dividirán en requisitos funcionales y requisitos no funcionales.

## Requisitos funcionales

**Identificador:** RE-F-1

**Descripción:** El acceso de los usuarios a la aplicación se realizará previa autenticación mediante un formulario de Login.

**Comentarios:** No se permite el acceso a usuarios que no estén autenticados en el sistema.

---

**Identificador:** RE-F-2

**Descripción:** El sistema tendrá dos perfiles de usuario: un usuario genérico y un usuario administrador.

**Comentarios:** El usuario administrador tendrá acceso a un número mayor de funcionalidades, concretamente la gestión de los usuarios (RE-F-3) y la gestión de la totalidad de los contenidos subidos (RE-F-4).

---

**Identificador:** RE-F-3

**Descripción:** El usuario administrador será el encargado de realizar la gestión de los usuarios: alta, baja y modificación de usuarios.

**Comentarios:** Esta funcionalidad no está permitida al usuario genérico.

---

**Identificador:** RE-F-4

**Descripción:** El sistema permitirá la gestión de los contenidos del repositorio: alta, baja y modificación de los contenidos.

**Comentarios:** El usuario administrador podrá gestionar todos los contenidos del sistema, mientras que el usuario genérico exclusivamente podrá gestionar los contenidos subidos por él.

---

**Identificador:** RE-F-5

**Descripción:** El sistema permitirá subir contenidos que se ajusten a un determinado estándar de visualización y entrega a elegir entre IMS LD, IMS QTI o ADL SCORM.

**Comentarios:** El estándar elegido, tras el estudio previo, será IMS QTI.

---

**Identificador:** RE-F-6

**Descripción:** Los contenidos subidos serán ficheros en formato zip que contendrán un fichero qti.xml con el contenido en cuestión y una serie de recursos que lo complementan.

**Comentarios:** Los recursos pueden ser ficheros HTML, imágenes, ficheros de audio, etc.

---

**Identificador:** RE-F-7

**Descripción:** El sistema permitirá subir descriptores que identifiquen el contenido con una serie de parámetros que describen ese contenido.

**Comentarios:** El descriptor será un fichero que cumpla el estándar IMS Learning Resource Meta-Data.

---

**Identificador:** RE-F-8

**Descripción:** El sistema permitirá la creación de descriptores para identificar a los contenidos subidos.

**Comentarios:** El descriptor creado contendrá los mismos parámetros que un descriptor subido al sistema.

---

**Identificador:** RE-F-9

**Descripción:** El sistema permitirá realizar búsquedas de contenidos a partir de un conjunto de parámetros de búsqueda.

**Comentarios:** La búsqueda recuperará los contenidos que más se ajusten a la descripción basada en los parámetros de búsqueda. Estos parámetros de búsqueda estarán basados en los descriptores de los contenidos.

---

**Identificador:** RE-F-10

**Descripción:** El sistema permitirá la descarga de los contenidos subidos.

---

**Identificador:** RE-F-11

**Descripción:** El sistema permitirá la visualización de un contenido de acuerdo con el estándar de presentación y entrega elegido.

**Comentarios:** El estándar elegido se especifica en el requisito RE-F-5.

---

**Identificador:** RE-F-12

**Descripción:** El sistema permitirá la composición de un material o contenido a partir de otros contenidos almacenados en el sistema. Como resultado de esta composición, el material nuevo incluirá un fichero imsmanifest.xml con la descripción de los recursos utilizados por el nuevo material.

**Comentarios:** El estándar elegido para la composición es el IMS Content Packaging.

## **Requisitos No Funcionales**

**Identificador:** RE-NF-1

**Descripción:** El sistema gestor de base de datos a utilizar será MySQL 5.0.

---

**Identificador:** RE-NF-2

**Descripción:** Los ficheros subidos al sistema se almacenarán en la propia base de datos.

---

**Identificador:** RE-NF-3

**Descripción:** La aplicación estará basada en una aplicación Web J2EE versión 1.3, con soporte para Struts 1.3.x.

---

**Identificador:** RE-NF-4

**Descripción:** El contenedor Web utilizado será Jakarta Tomcat 5.5.20

#### 4.1.2 Casos de uso

Una vez que los requisitos han quedado claros, a continuación se presentan los diagramas de casos de uso, junto con los diagramas de actividad y casos de uso extendidos, para describir el comportamiento del sistema de una manera más detallada.

Por cada caso de uso, se muestra su diagrama, su caso de uso extendido y su diagrama de actividad, que demuestra la serie de actividades que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en el caso de uso. Existen casos de uso que están formados por varios diagramas de actividad y varios casos de uso extendidos.

- Caso de uso “Login”

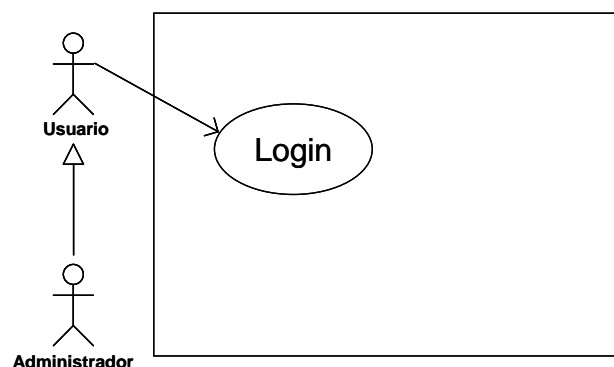


Figura 4.1. Diagrama caso de uso Login

El caso de uso “Login” está relacionado con los siguientes requisitos: RE-F-1 y RE-F-2.

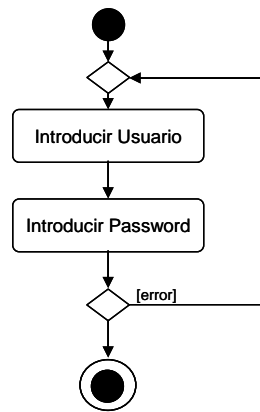


Figura 4.2. Diagrama de actividad del caso de uso Login

Caso de uso extendido:

<b>Caso de Uso</b>	Login
<b>Actores</b>	Usuario
<b>Objetivo</b>	Validar al usuario en el sistema
<b>Precondiciones</b>	Usuario existente en el sistema
<b>Postcondiciones</b>	Usuario validado en el sistema
<b>Escenario básico</b>	Introducir Login Introducir Password Verificar Login/Password Usuario validado

▪ Caso de uso “Administrar repositorio”

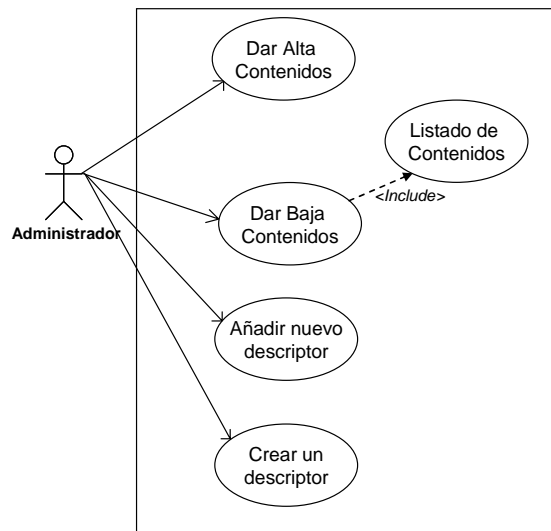


Figura 4.3. Diagrama del caso de uso Administrar repositorio

El caso de uso "Administrar repositorio" está relacionado con los siguientes requisitos: RE-F-4, RE-F-5, RE-F-6, RE-F-7 y RE-F-8.

Para este caso de uso, se incluyen cinco diagramas de actividad, para mostrar la actividad de los casos de uso contenidos en "Administrar repositorio".

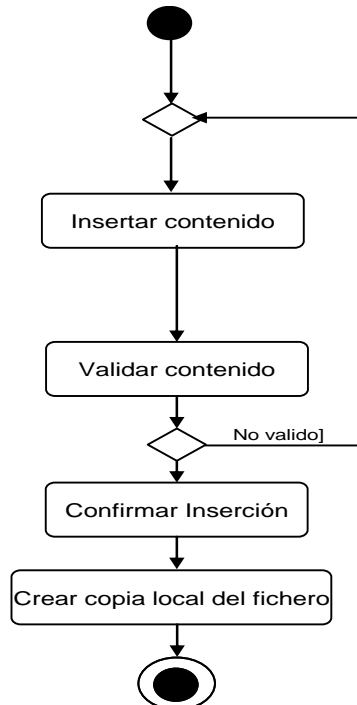


Figura 4.4. Diagrama de actividad de Alta contenido

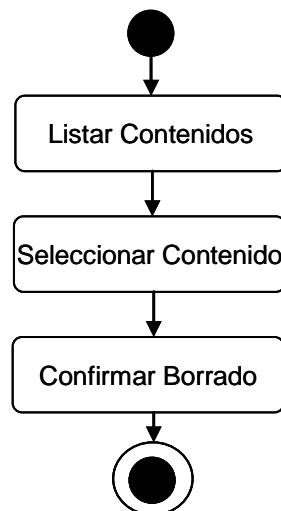


Figura 4.5. Diagrama de actividad del caso de uso Baja contenido



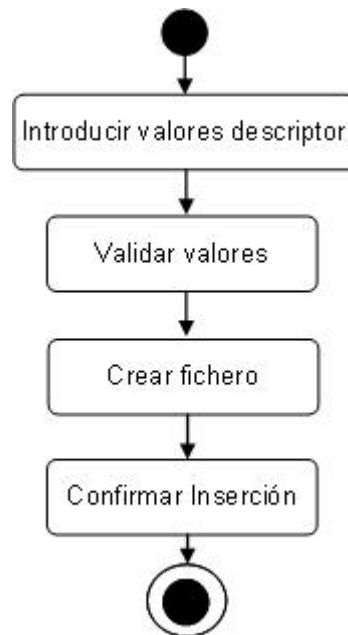


Figura 4.6. Diagrama de actividad del caso de uso Crear Descriptor

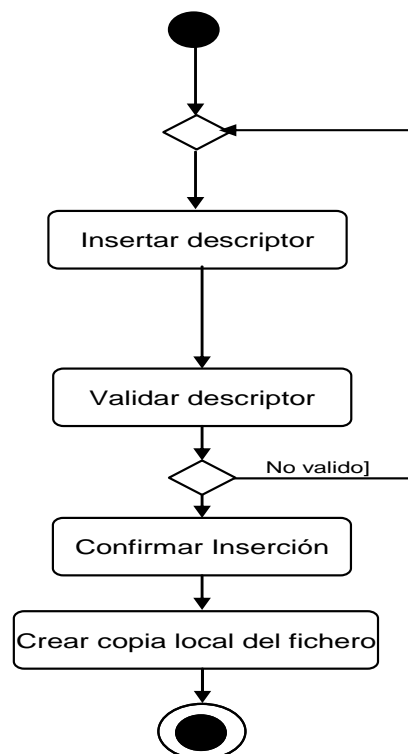


Figura 4.7. Diagrama de actividad del caso de uso Añadir Descriptor

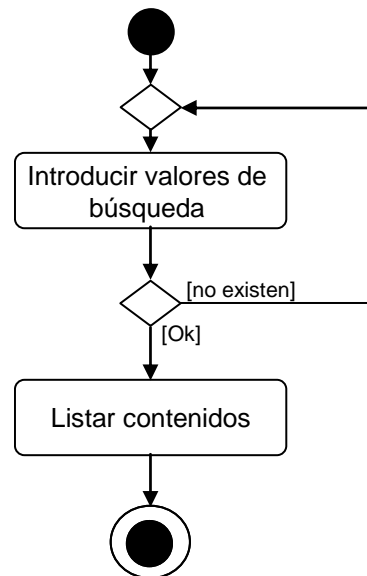


Figura 4.8. Diagrama de actividad del caso de uso Listado de contenidos

A continuación se presentan los casos de uso extendidos de los casos de uso que componen “Administrar repositorio”:

<b>Caso de Uso</b>	Dar Alta Contenidos
<b>Actores</b>	Usuario
<b>Objetivo</b>	Insertar un contenido en el repositorio
<b>Precondiciones</b>	Usuario logado en el sistema
<b>Poscondiciones</b>	Contenido incluido en el repositorio
<b>Escenario básico</b>	Introducir Nombre contenido Introducir el fichero del contenido Validar contenido Confirmar inserción en repositorio Guardar fichero en la base de datos

<b>Caso de Uso</b>	Dar de Baja Contenidos
<b>Actores</b>	Usuario
<b>Objetivo</b>	Borrar un contenido en el repositorio
<b>Precondiciones</b>	Usuario autenticado en el sistema
<b>Poscondiciones</b>	Contenido borrado en el repositorio
<b>Escenario básico</b>	Listar todos los contenidos creados por el usuario Marcado de contenidos a borrar Confirmar borrado en repositorio Borrar posibles descriptores aislados.

<b>Caso de Uso</b>	Añadir nuevo descriptor
<b>Actores</b>	Usuario
<b>Objetivo</b>	Insertar un descriptor en el repositorio
<b>Precondiciones</b>	Usuario autenticado en el sistema Contenido subido
<b>Poscondiciones</b>	Descriptor insertado en el repositorio
<b>Escenario básico</b>	Introducir fichero descriptor Validar descriptor Confirmar inserción en repositorio Almacenar fichero en la base de datos

<b>Caso de Uso</b>	Crear descriptor
<b>Actores</b>	Usuario
<b>Objetivo</b>	Insertar un descriptor creado por el usuario en el repositorio
<b>Precondiciones</b>	Usuario autenticado en el sistema
<b>Poscondiciones</b>	Descriptor insertado en el repositorio
<b>Escenario básico</b>	Introducir valores del descriptor. Crear fichero descriptor a partir de los valores introducidos. Confirmar inserción en repositorio.

<b>Caso de Uso</b>	Listado de Contenidos
<b>Actores</b>	Administrador
<b>Objetivo</b>	Mostrar los contenidos buscados del repositorio
<b>Precondiciones</b>	Usuario Administrador autenticado en el sistema
<b>Poscondiciones</b>	Se muestran los contenidos del sistema que cumplen con el patrón de búsqueda
<b>Escenario básico</b>	Buscar contenidos en el repositorio. Mostrar los resultados encontrados.

- Caso de uso “**Administrar usuarios**”

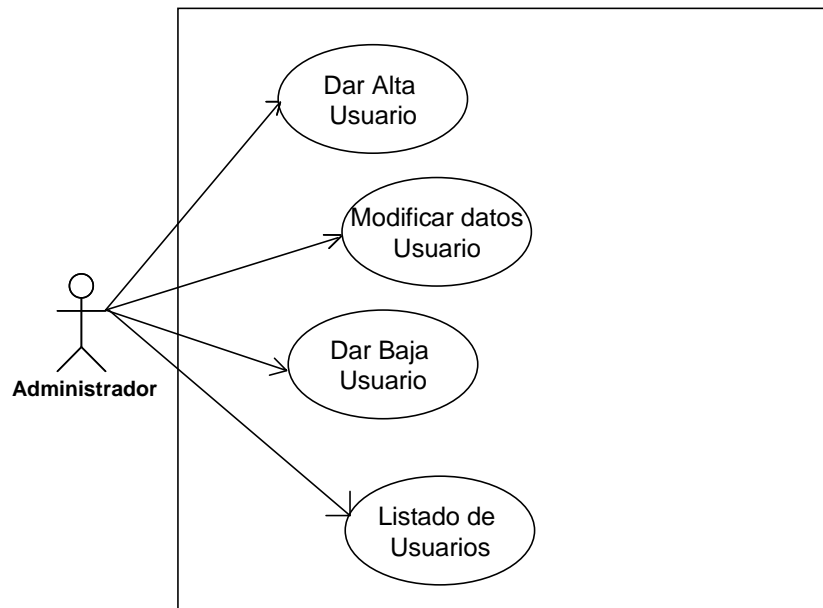


Figura 4.9. Diagrama del caso de uso Administrar usuarios

El caso de uso “Administrar usuarios” está relacionado con los siguientes requisitos: RE-F-3.

Para este caso de uso, se incluyen tres diagramas de actividad, para mostrar la actividad de los casos de uso contenidos en “Administrar usuarios”.

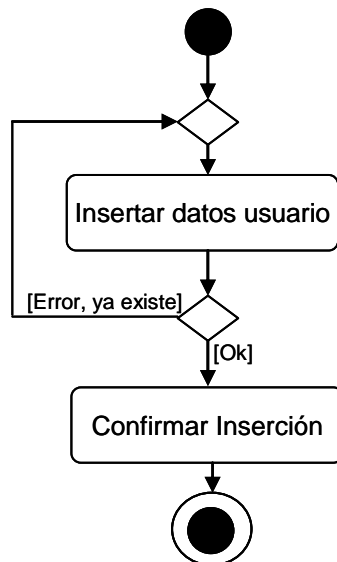


Figura 4.10. Diagrama de actividad del caso de uso Alta usuario

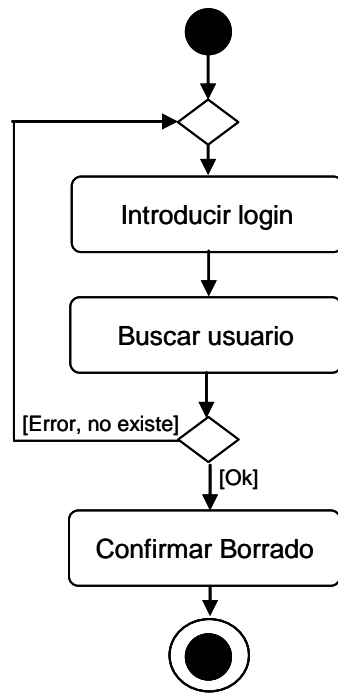


Figura 4.11. Diagrama de actividad del caso de uso Baja Usuario

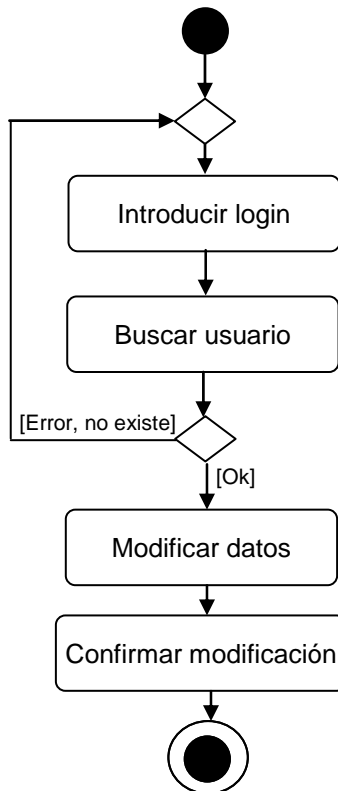


Figura 4.12. Diagrama de actividad del caso de uso Modificar Usuario



Figura 4.13. Diagrama de actividad del caso de uso Listar Usuarios

A continuación se muestran los casos de uso extendidos asociados al caso de uso “Administrar usuarios”.

<b>Caso de Uso</b>	Dar Alta Usuario
<b>Actores</b>	Administrador
<b>Objetivo</b>	Crear un nuevo usuario del repositorio y del sistema
<b>Precondiciones</b>	Usuario Administrador autenticado en el sistema
<b>Poscondiciones</b>	Creado nuevo usuario del repositorio
<b>Escenario básico</b>	Insertar datos del nuevo usuario Confirmar creación del usuario

<b>Caso de Uso</b>	Dar Baja Usuario
<b>Actores</b>	Administrador
<b>Objetivo</b>	Eliminar un usuario del repositorio
<b>Precondiciones</b>	Usuario Administrador autenticado en el sistema El usuario a eliminar existe en el sistema
<b>Poscondiciones</b>	Usuario eliminado del sistema
<b>Escenario básico</b>	Introducir login de usuario a borrar Buscar usuario a borrar Confirmar borrado

Caso de Uso	Modificar Datos Usuario
Actores	Administrador
Objetivo	Modificar uno o más datos de un usuario
Precondiciones	Usuario Administrador logado en el sistema El usuario a modificar existe en el sistema
Poscondiciones	Se guardan los nuevos datos del usuario
Escenario básico	Introducir login del usuario a modificar Buscar usuario en el sistema Modificar datos del usuario Confirmar modificación

- Caso de uso “Realizar Búsquedas”

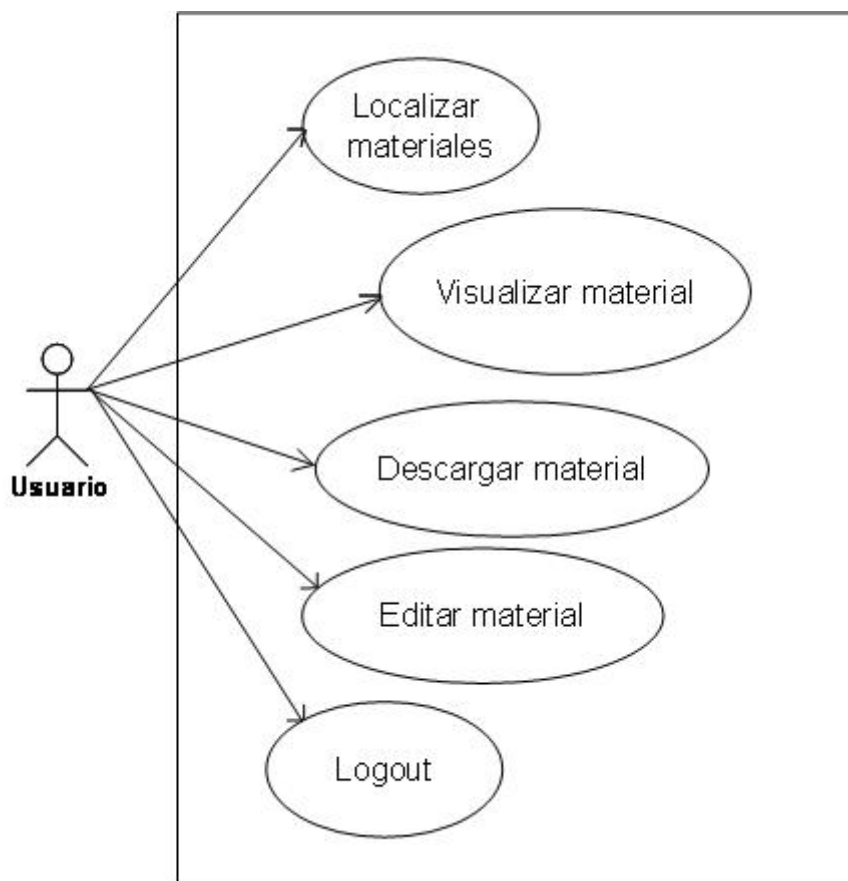


Figura 4.14. Diagrama de caso de uso de Realizar Búsquedas

El caso de uso “Realizar Búsquedas” está relacionado con los siguientes requisitos: RE-F-9, RE-F-10 y RE-F-11.

Para este caso de uso, se incluyen tres diagramas de actividad, para mostrar la actividad de los casos de uso contenidos en "Realizar Búsquedas".

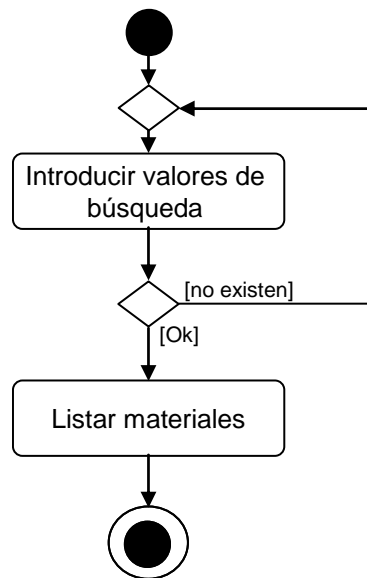


Figura 4.15. Diagrama de actividad del caso de uso Localizar Materiales

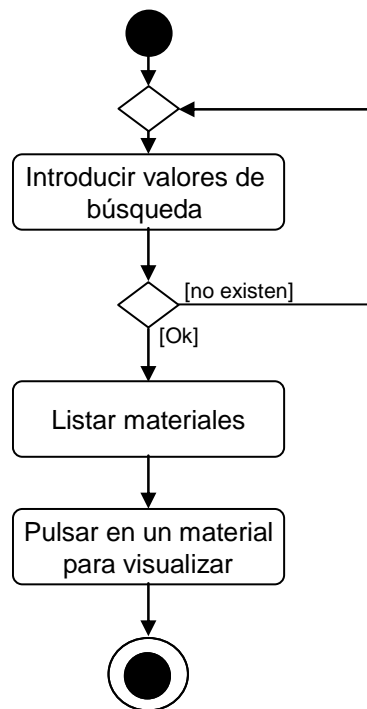


Figura 4.16. Diagrama de actividad del caso de uso Visualizar material



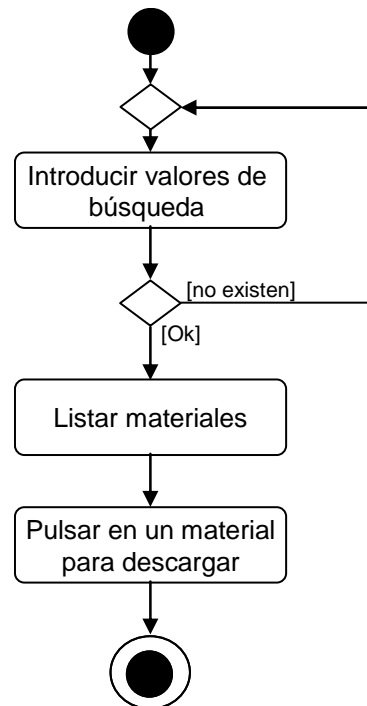


Figura 4.17. Diagrama de actividad del caso de uso Descargar material

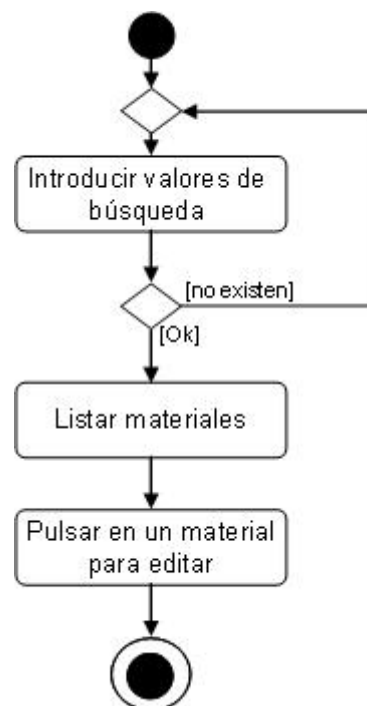


Figura 4.18. Diagrama de actividad del caso de uso Editar material

A continuación se muestran los casos de uso extendidos asociados al caso de uso "Realizar Búsquedas".

<b>Caso de Uso</b>	Localizar materiales
<b>Actores</b>	Usuario
<b>Objetivo</b>	Buscar materiales en el repositorio
<b>Precondiciones</b>	Usuario autenticado en el sistema
<b>Poscondiciones</b>	Se muestra el Listado de materiales que cumple con los criterios especificados
<b>Escenario básico</b>	Introducir valores de búsqueda. Listar materiales.

<b>Caso de Uso</b>	Visualizar un material
<b>Actores</b>	Usuario
<b>Objetivo</b>	Buscar y visualizar un material
<b>Precondiciones</b>	Usuario logado en el sistema
<b>Poscondiciones</b>	Material visualizado
<b>Escenario básico</b>	Introducir valores de búsqueda. Listar materiales. Pulsar en un material para visualizarlo.

<b>Caso de Uso</b>	Descargar un material
<b>Actores</b>	Usuario
<b>Objetivo</b>	Buscar y descargar un material
<b>Precondiciones</b>	Usuario logado en el sistema
<b>Poscondiciones</b>	Se puede descargar el Material
<b>Escenario básico</b>	Introducir valores de búsqueda. Listar materiales. Pulsar en un material para descargarlo.

<b>Caso de Uso</b>	Editar un material
<b>Actores</b>	Usuario
<b>Objetivo</b>	Buscar y editar un material
<b>Precondiciones</b>	Usuario logado en el sistema
<b>Poscondiciones</b>	Se puede editar el Material
<b>Escenario básico</b>	Introducir valores de búsqueda. Listar materiales. Pulsar en un material para editarlo.

- Caso de uso “Realizar composiciones”

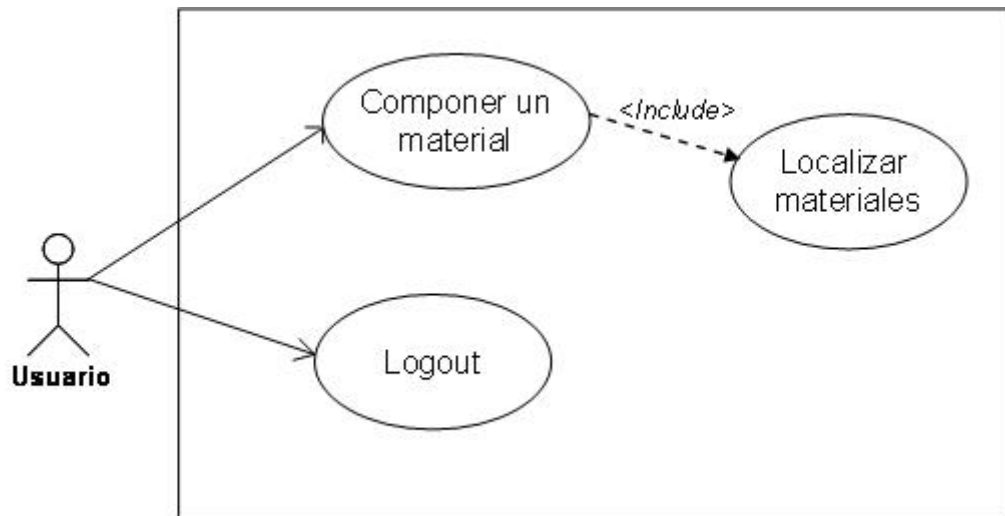


Figura 4.19. Diagrama del caso de uso Realizar composiciones

El caso de uso “Realizar composiciones” está relacionado con los siguientes requisitos: RE-F-9 y RE-F-12.

Para este caso de uso, se incluyen dos diagramas de actividad, para mostrar la actividad de los casos de uso contenidos en “Realizar composiciones”, ya que el caso de uso “Localizar materiales” ya ha sido mostrado en el anterior caso de uso.

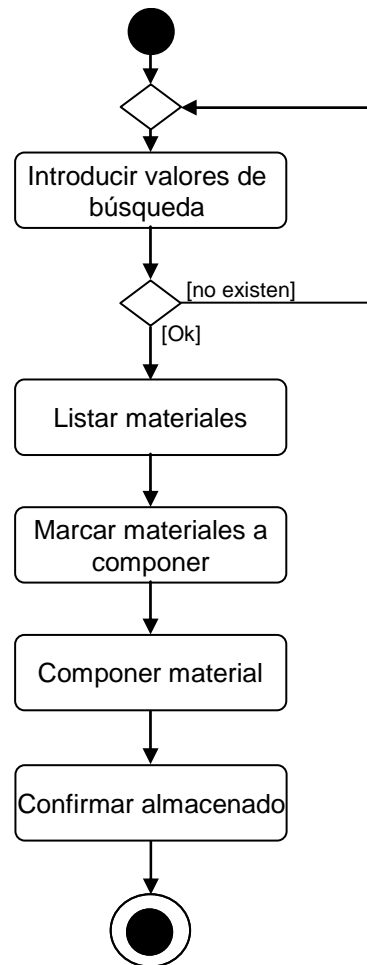


Figura 4.20. Diagrama de actividad del caso de uso Componer un material

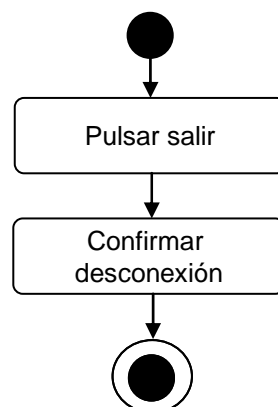


Figura 4.21. Diagrama de actividad del caso de uso Logout

A continuación se muestran los casos de uso extendidos asociados al caso de uso "Realizar composiciones".

<b>Caso de Uso</b>	Componer un material
<b>Actores</b>	Usuario
<b>Objetivo</b>	Buscar contenidos y componer un material
<b>Precondiciones</b>	Usuario logado en el sistema
<b>Poscondiciones</b>	Composición de un nuevo material y éste se guarda en el repositorio
<b>Escenario básico</b>	Introducir valores de búsqueda. Listar materiales. Seleccionar materiales destinados a la composición. Componer material. Confirmar almacenado en el repositorio.

<b>Caso de Uso</b>	Logout
<b>Actores</b>	Usuario
<b>Objetivo</b>	Salir de la aplicación
<b>Precondiciones</b>	Usuario logado en el sistema
<b>Poscondiciones</b>	Usuario desconectado de la aplicación
<b>Escenario básico</b>	Pulsar sobre la opción de salir de la aplicación. Confirmar desconexión.

## 4.2 Diseño

Durante esta fase del trabajo se realizó el diseño de la aplicación. Se parte del análisis detallado realizado en la anterior fase sobre la funcionalidad de la herramienta y se realizan los diseños de la capa de persistencia, de la capa de lógica de negocio y de la interfaz gráfica.

#### 4.2.1 Diseño de la capa de persistencia

Para el diseño de la capa de persistencia se han tenido en cuenta las entidades lógicas necesarias para el correcto almacenamiento de la información. En la Figura 4.23 se muestra el diagrama entidad/relación correspondiente a la disposición de la información gestionada por el sistema.

Las entidades que componen el esquema Entidad/Relación y sobre las que se van a realizar las operaciones son cuatro:

- **Usuarios.** La entidad lógica que almacena la información asociada a los usuarios del sistema.
- **Items.** La entidad lógica que almacena la información de los contenidos educativos que va a albergar el sistema.
- **Descriptores.** La entidad lógica que almacena las características que describen a los contenidos educativos que gestiona el sistema.
- **Materiales.** La entidad lógica que almacena la información a los materiales que han sido compuestos a partir de contenidos educativos, que ya estaban almacenados en el sistema.

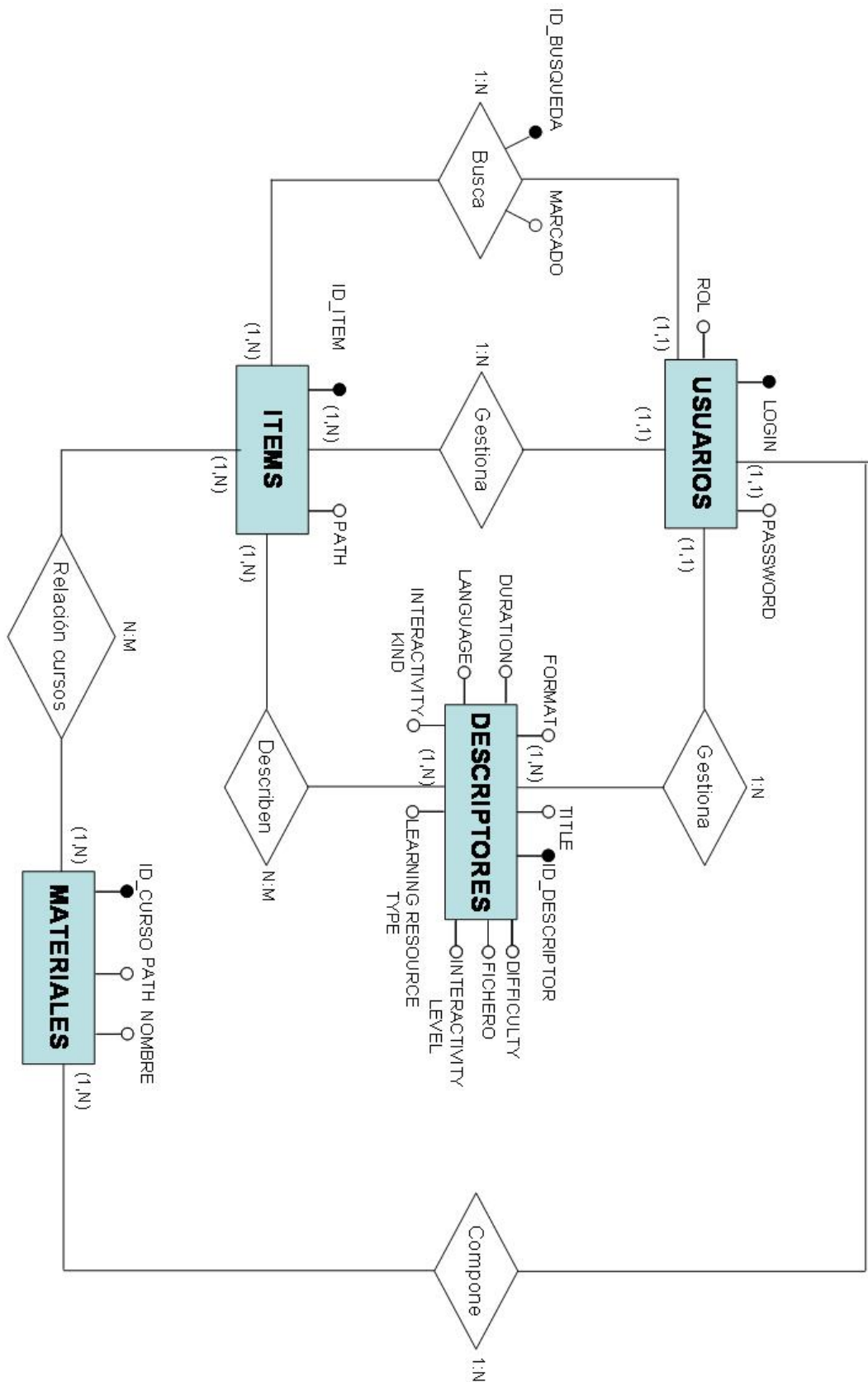


Figura 4.22. Esquema Entidad/Relación de la base de datos

En la Figura 4.23 se muestra el esquema que muestra el Modelo Relacional asociado al esquema Entidad/Relación anterior. En este esquema se puede ver con mayor claridad la transformación de las entidades y de las relaciones a tablas de la base de datos. Esto se puede apreciar también en el script de creación de la BBDD incluido en el apartado de Configuración del sistema del manual de usuario que se encuentra en el capítulo 6.



Modelo Relacional

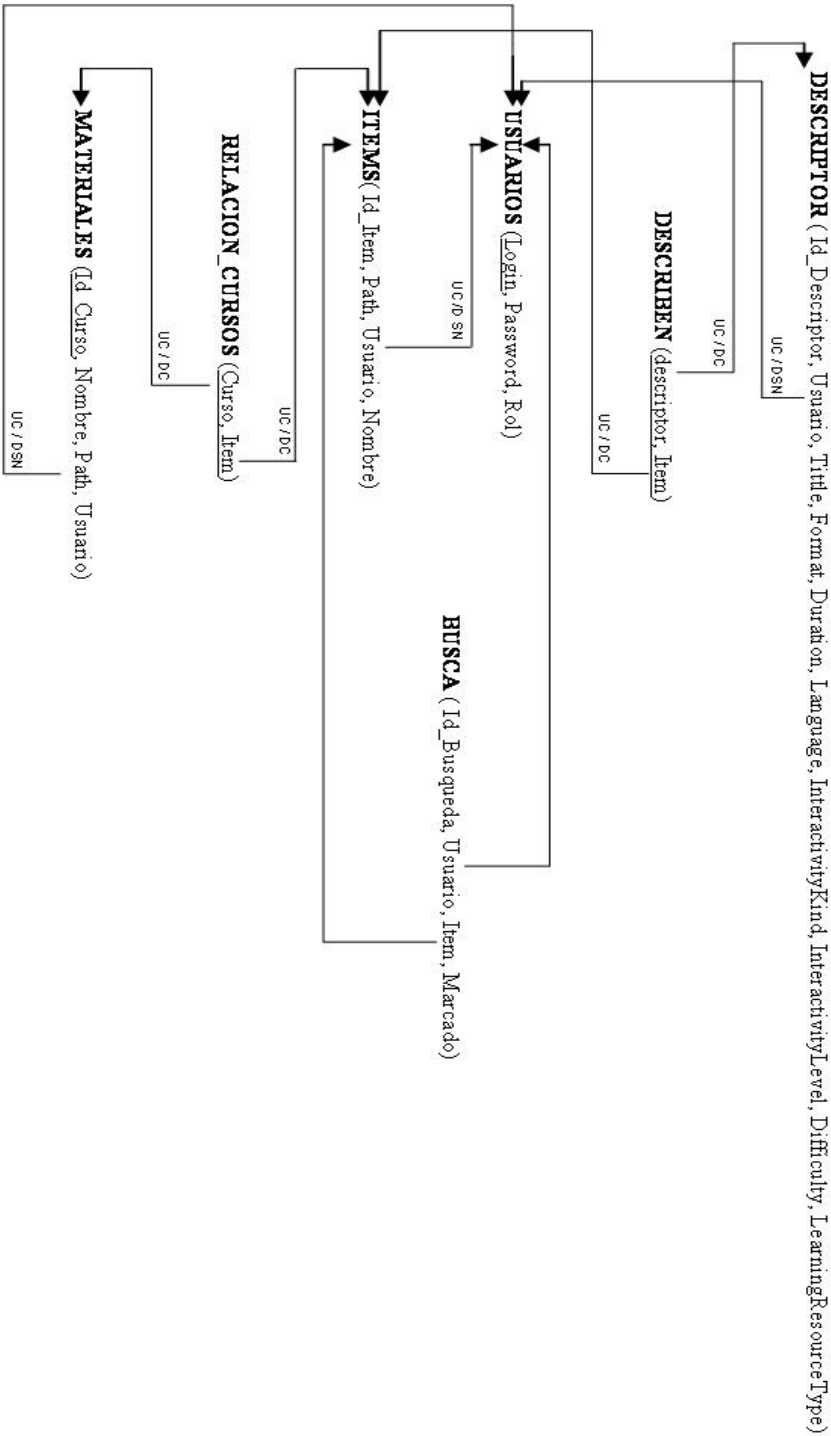


Figura 4.23. Esquema del modelo Relacional de la base de datos

#### 4.2.2 Diseño de la capa de lógica de negocio

El diseño de la lógica de negocio de la aplicación está basado en el modelo de clases que se puede ver en la Figura 4.24. Un **modelo o diagrama de clases** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

El modelo de clases de la aplicación consta de cinco clases que representan las entidades lógicas de la capa de negocio, cuyo detalle es:

- **Usuario.** La clase Usuario representa al usuario de la aplicación y contiene las propiedades que lo van a definir, que son el login, la password y el rol del usuario, así como la funcionalidad asociada a dicho usuario.
- **Administrador.** La clase Administrador hereda de la clase Usuario y añade funcionalidad para la gestión de los usuarios de la aplicación.
- **Contenido.** La clase Contenido representa los contenidos educativos que se van a gestionar en el sistema. La funcionalidad principal de esta clase es la gestión de los contenidos, así como la descarga y visualización de los mismos.
- **Material.** La clase Material representa los contenidos educativos que han sido compuestos a partir de otros que ya estaban almacenados en el sistema. Su principal funcionalidad es la composición de dichos materiales.
- **Descriptor.** La clase Descriptor es una entidad lógica que engloba lo que se conoce como descriptores, que en el caso del modelo se corresponden con las propiedades de la clase, es decir, cada propiedad es un descriptor del contenido al que está asociado. La funcionalidad de esta clase es la correspondiente a la gestión de los descriptores de los contenidos.

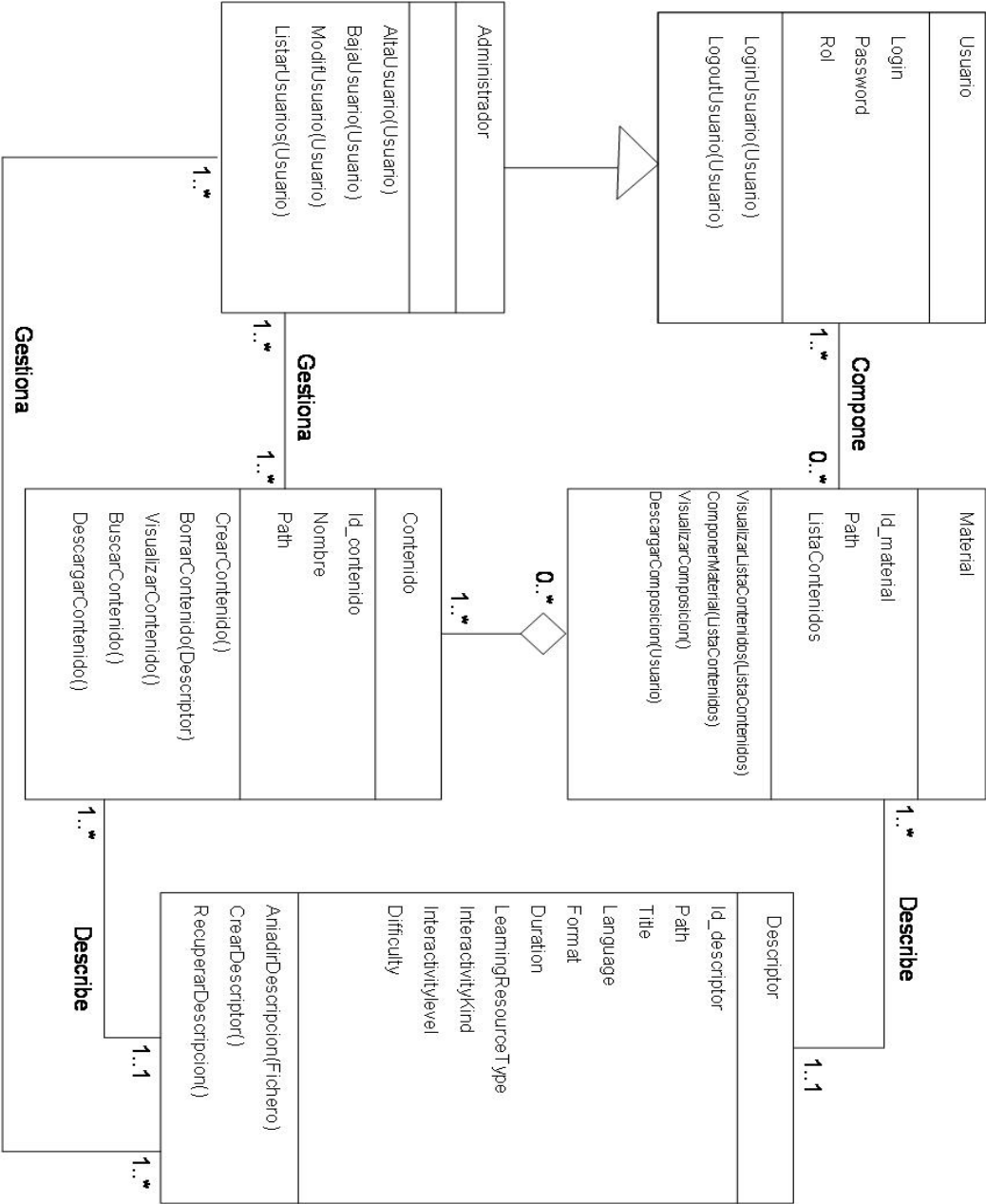


Figura 4.24. Modelo de clases del sistema

Una vez que la parte estática del sistema ha quedado modelada, es necesario modelar los aspectos dinámicos del sistema, a través de los **diagramas de interacción**, lo que conlleva modelar instancias concretas de clases, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. De esta manera se consigue describir la forma en que grupos de objetos colaboran para proveer un comportamiento.

Para describir el comportamiento dinámico del sistema son necesarios dos tipos de diagramas de interacción: el diagrama de secuencia y los diagramas de colaboración.

El diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes. En la Figura 4.25 se puede ver el diagrama de secuencia del sistema.

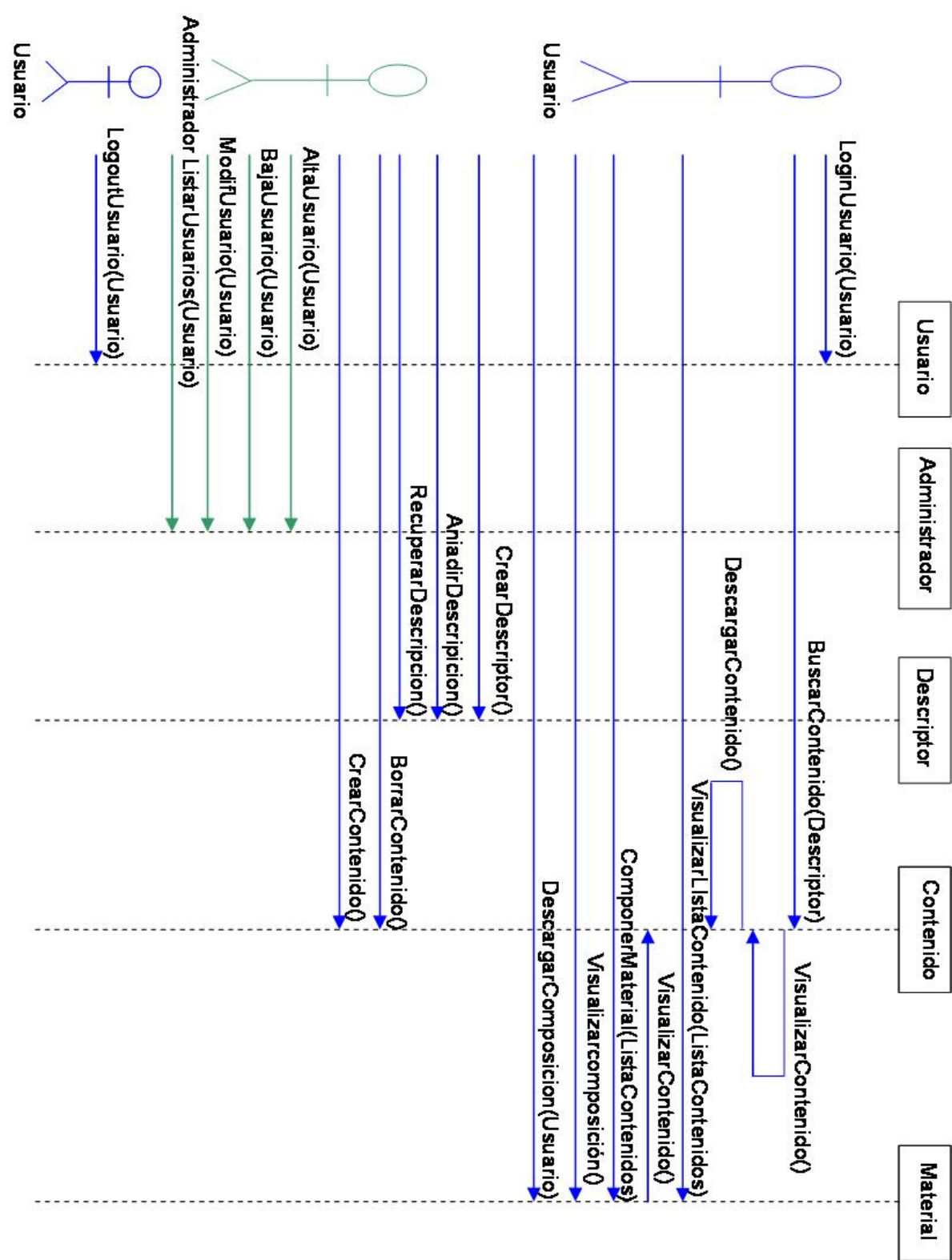


Figura 4.25. Diagrama de secuencia del sistema

Los diagramas de colaboración permiten mostrar las interacciones entre objetos organizadas entorno a los objetos y los enlaces entre ellos, para ello se realiza un diagrama de colaboración por cada caso de uso en el que sea relevante mostrar la colaboración entre los objetos que forman parte de ese caso de uso.

Las siguientes figuras muestran los distintos diagramas de colaboración que se corresponden con casos de uso explicados en el apartado de análisis.

- Diagrama de colaboración correspondiente a los casos de uso de Login y Logout.

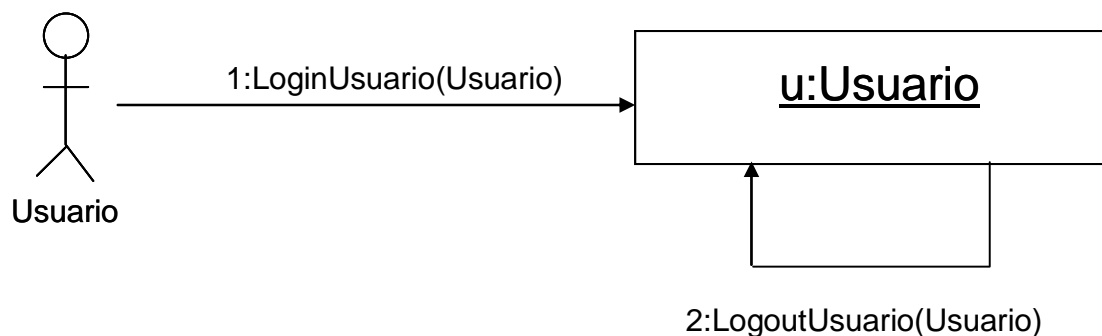


Figura 4.26. Diagrama de colaboración de los casos de uso Login y Logout

- Diagrama de colaboración correspondiente al caso de uso Administrar Usuarios.

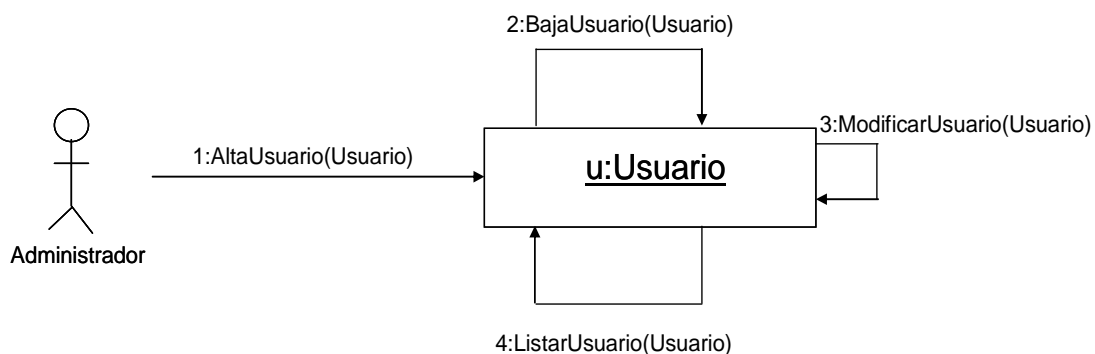


Figura 4.27. Diagrama de colaboración del caso de uso Administrar Usuarios

- Diagrama de colaboración correspondiente al caso de uso Realizar Búsquedas.

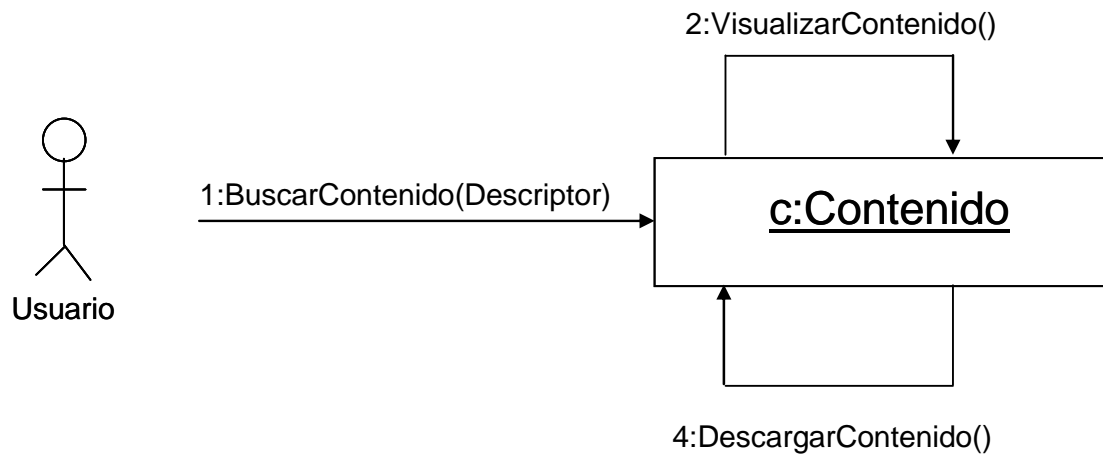


Figura 4.28. Diagrama de colaboración del caso de uso Realizar Búsquedas

- Diagrama de colaboración correspondiente al caso de uso Realizar composiciones.

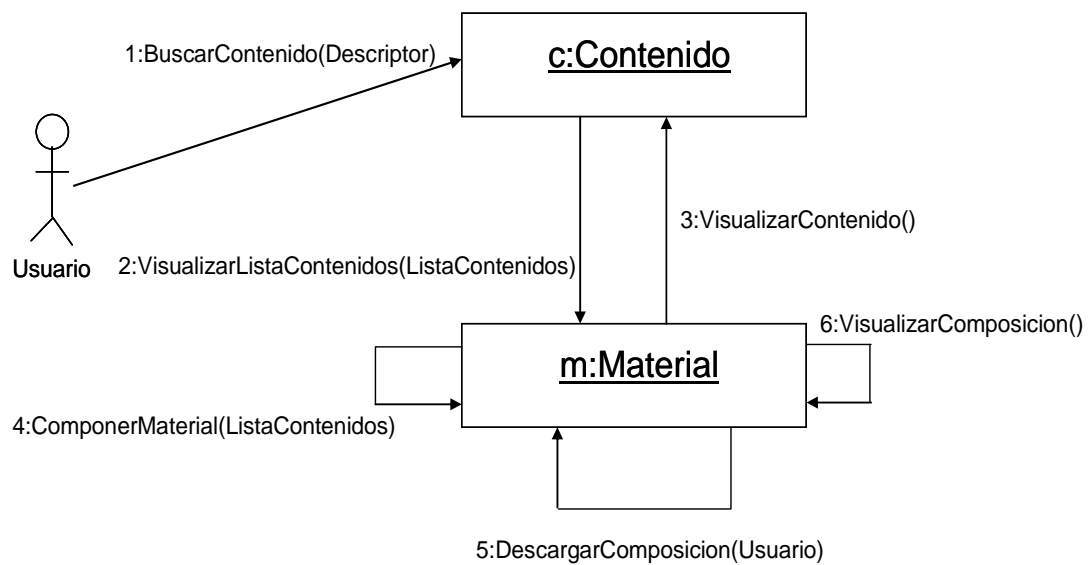


Figura 4.29. Diagrama de colaboración del caso de uso Realizar composiciones

- Diagrama de colaboración correspondiente al caso de uso Administrar Repositorio.

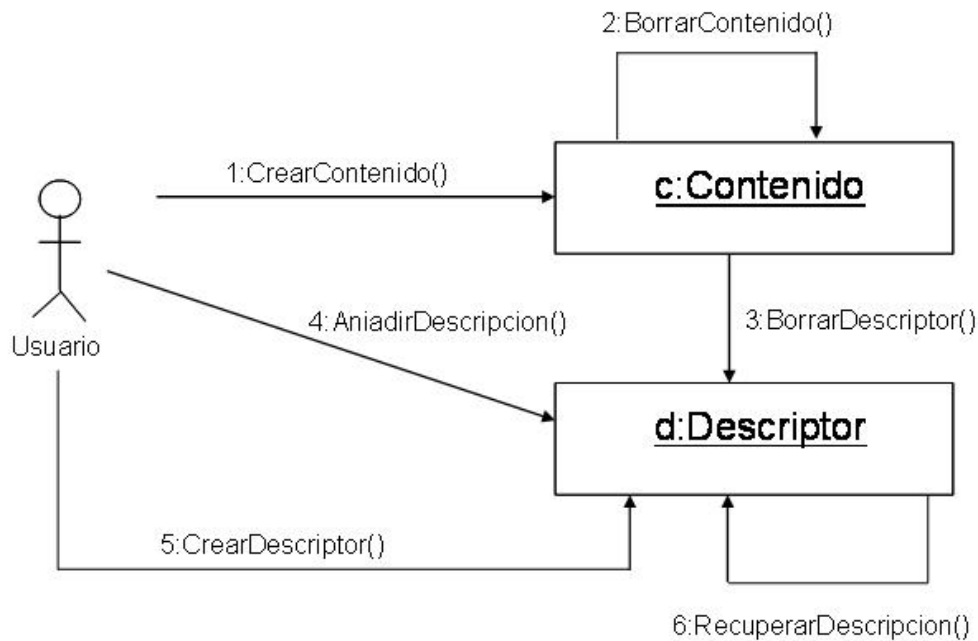


Figura 4.30. Diagrama de colaboración del caso de uso Administrar Repositorio

#### 4.2.3 Diseño de la interfaz gráfica

La interfaz gráfica de esta aplicación está formada por un conjunto de páginas JSP. El diseño de estas páginas se realizó teniendo en cuenta los estándares del W3C, XHTML 1.0 y CSS.

Los objetivos de este diseño eran conseguir una interfaz gráfica con un grado aceptable de accesibilidad y compatibilidad entre navegadores, además de ser agradable para el usuario. Las páginas están diseñadas dividiendo su espacio en cuatro partes: un encabezado, en el que aparece el nombre de la aplicación y enlaces generales a todo el sistema, un margen izquierdo que contendrá enlaces específicos de la funcionalidad actual de cada página, una parte central, en la que se mostrarán los distintos formularios de datos y otra información y, por último, un pie de página que mostrará información relativa a los autores de la herramienta. La utilización de hojas de estilo CSS permite variar de forma rápida y sencilla la apariencia de la interfaz gráfica de esta aplicación separando contenido y presentación.

En la Figura 4.31 se muestra un ejemplo de la interfaz. Para que quede más claro la división de las cuatro partes, se han añadido unas líneas rojas que facilitan distinguir mejor las partes que componen el diseño de la interfaz.



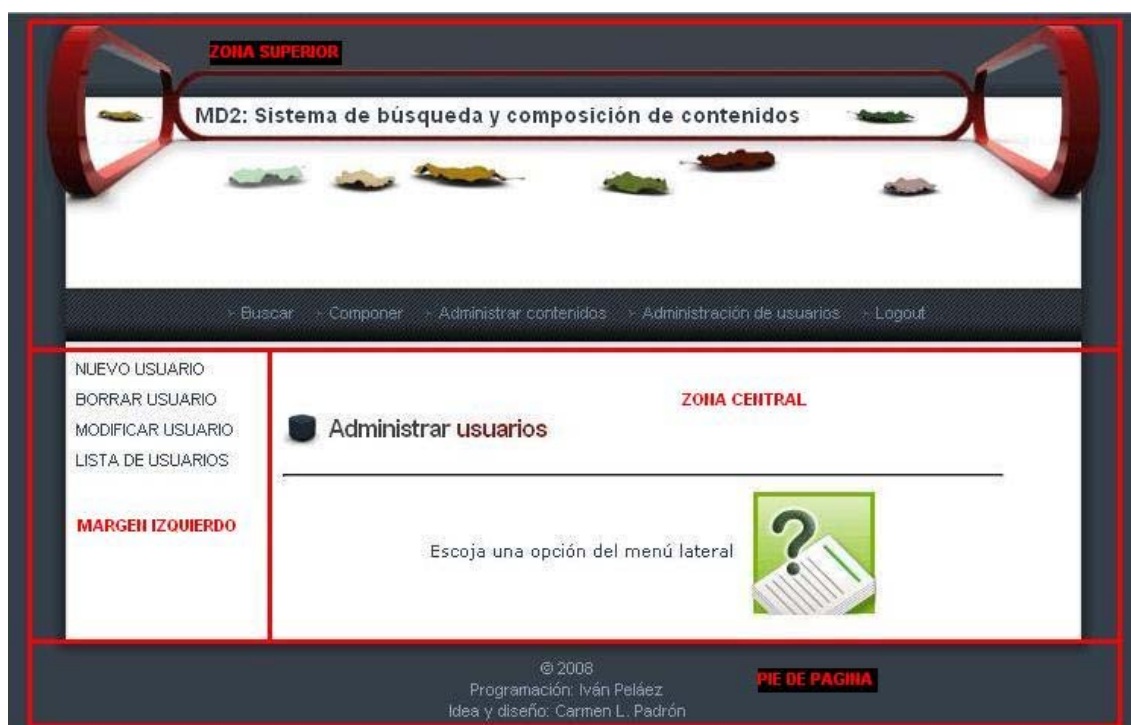


Figura 4.31. Interfaz de ejemplo donde se observan las partes explicadas

---

## 5. MANUAL DE USUARIO

---

Este manual intenta ofrecer al usuario un conjunto de instrucciones acerca del manejo de la aplicación. A lo largo de este manual, se irá describiendo en detalle la secuencia de pasos que deben realizarse para cada una de las funcionalidades de las que consta la herramienta. Y al final del capítulo se ofrecen las instrucciones para configurar la aplicación.

### *5.1. Acceso a la aplicación y menú principal*

La herramienta es una aplicación web, por ello, debe estar instalada en algún servidor. Para acceder a ella es necesario abrir un navegador de Internet. Desde este programa se accede a la página de bienvenida de la aplicación visitando la url:

<http://md2.dei.inf.uc3m.es:8080/qticomposer>

La Figura 5.1 presenta la página de bienvenida del sistema. Esta página es un formulario de *Login* en el que el usuario debe autenticarse, para ello debe introducir su *login* y su *password*. Estos datos le han sido proporcionados previamente por el administrador del sistema.

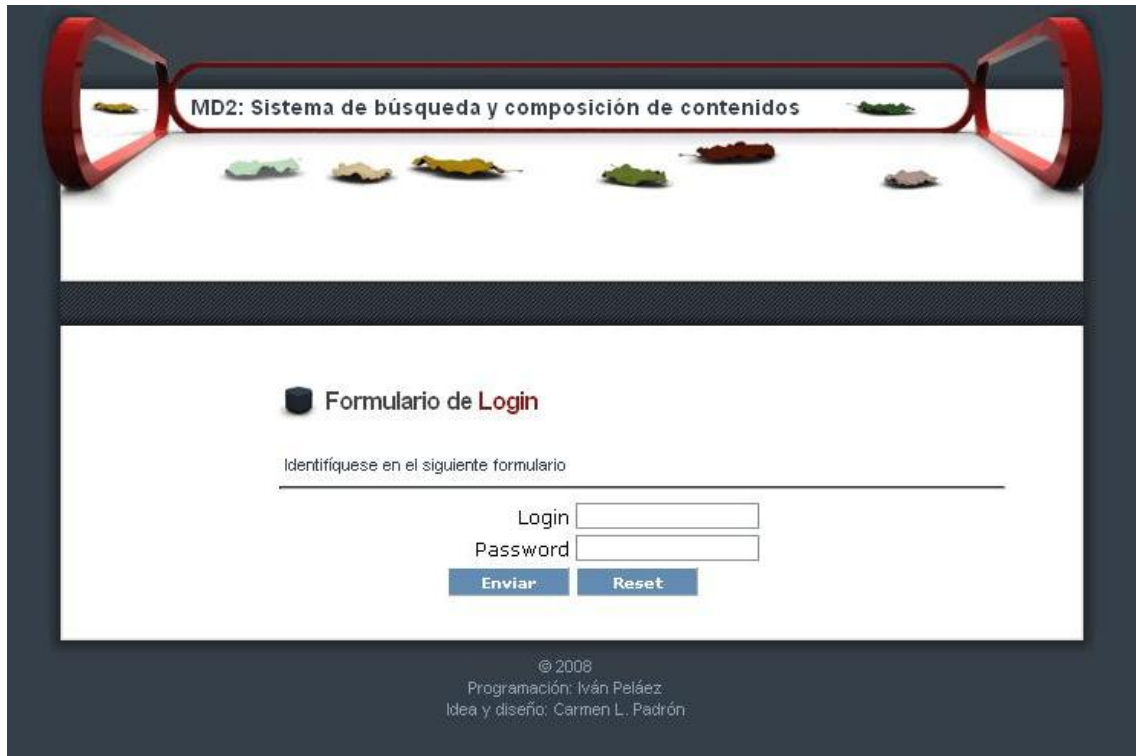


Figura 5.1. Formulario de login de la aplicación

Una vez que el usuario se haya logado correctamente se cargará la página del menú principal del sistema. Esta página variará dependiendo del rol del usuario, ya que si el usuario es Administrador dispondrá de más funcionalidades, de acuerdo con el diseño de la aplicación, explicado en los capítulos anteriores. En la Figura 5.2 podemos ver la página principal de un usuario Administrador. La única funcionalidad a la que no puede acceder un usuario Genérico es a la administración de usuarios.

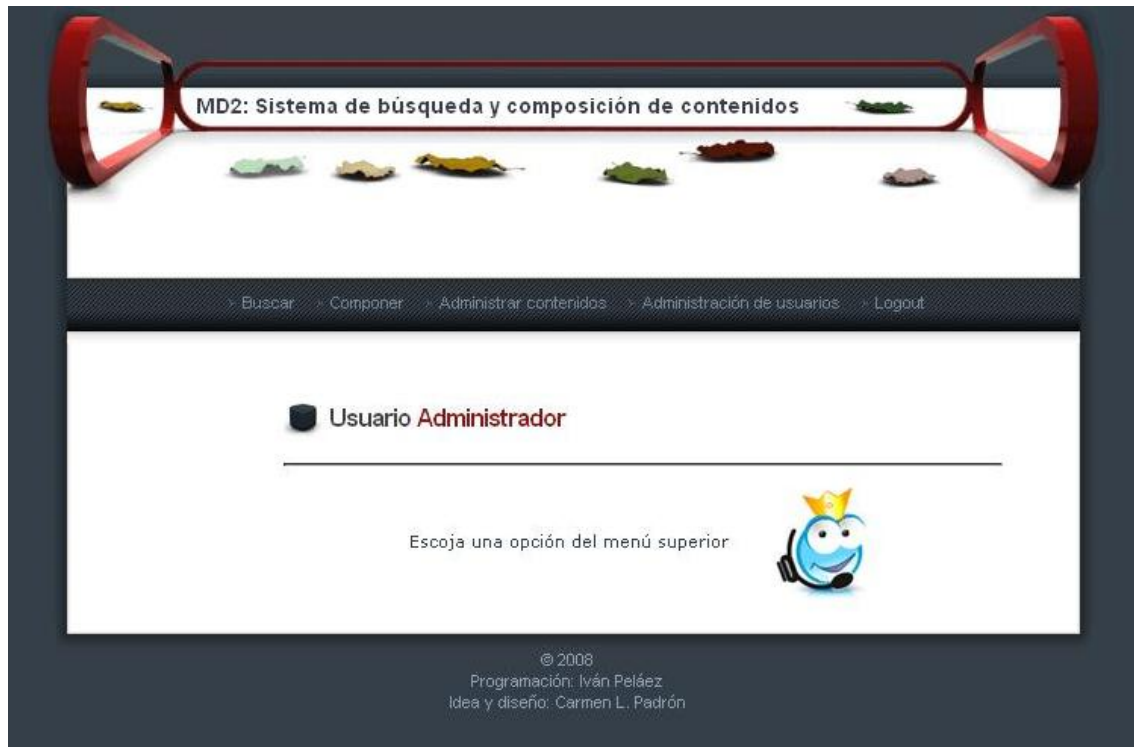


Figura 5.2. Página principal del usuario Administrador

Esta página está dividida en cuatro partes, y sigue un patrón de diseño común a toda la aplicación, en la parte superior se encuentra un encabezado con el nombre de la aplicación, debajo de este encabezado se sitúa el menú principal que será visible en todas las partes del sistema, en el lateral izquierdo se sitúa una zona de menús cuyas opciones variarán dependiendo del contenido donde nos encontremos. En la zona central aparecerán los datos y formularios necesarios para el correcto funcionamiento de la aplicación y por último en la zona inferior se encuentran datos relativos a los autores de la herramienta, y que también son comunes a todo el sistema.

Las opciones del menú principal, y que son comunes a todo el sistema, son las siguientes:

- **Buscar.** Con esta opción nos dirigimos al buscador de la aplicación para localizar contenidos almacenados.
- **Componer.** Con esta opción nos dirigimos a un buscador de contenidos para efectuar una búsqueda de contenidos, de los cuales podremos escoger varios para componer un contenido nuevo.
- **Administrar Contenidos.** Con esta opción podremos subir tanto contenidos nuevos como descriptores de dichos contenidos.
- **Administrar Usuarios.** Esta opción solo es visible si el usuario tiene rol Administrador y permite la gestión de los usuarios.

- Logout. Con esta opción salimos de la aplicación volviendo al formulario de identificación o *Login*.

## 6.2. Administrar Usuarios

Esta parte de la aplicación sólo es accesible para usuarios administradores y permite la gestión de los usuarios registrados en el sistema. Como se puede ver en la Figura 5.3. en el menú lateral izquierdo se distinguen cuatro opciones con funcionalidad distinta, éstas son la creación de un nuevo usuario, la eliminación de un usuario, la modificación de los datos de un usuario y la posibilidad de listar todos los usuarios almacenados en el sistema.



Figura 5.3. Página para la administración de usuarios

Para crear un usuario pulsamos en la opción correspondiente del menú lateral izquierdo "NUEVO USUARIO", esta acción nos llevará a la siguiente pantalla donde podremos introducir los datos del nuevo usuario. Como muestra la Figura 5.4. los datos a introducir son el *login* o identificador con el que el usuario se registrará en el sistema, la password correspondiente al usuario para dotar de seguridad al sistema y el rol del usuario; que podrá ser rol Administrador o rol Genérico.

The screenshot displays the MD2 system interface. At the top, a red 3D frame contains the title "MD2: Sistema de búsqueda y composición de contenidos". Below the title, a navigation bar includes links: "Buscar", "Componer", "Administrar contenidos", "Administración de usuarios", and "Logout". The main content area features a "VOLVER" link and a section titled "Nuevo usuario" with a small icon. Below this title, a prompt reads "Introduzca los datos del nuevo usuario". The form contains three input fields: "Login", "Password", and "Rol" (a dropdown menu currently showing "Genérico"). At the bottom of the form are two buttons: "Aceptar" and "Reset". The footer of the page includes copyright information: "© 2008", "Programación: Iván Peláez", and "Idea y diseño: Carmen L. Padrón".

Figura 5.4. Formulario de nuevo usuario

Tras introducir los datos, pulsamos en el botón de “Aceptar” y si no se produce ningún fallo, el usuario se almacenará en el repositorio del sistema, y la aplicación nos confirmará que el usuario se ha almacenado correctamente, como muestra la Figura 5.5.



Figura 5.5. Página de usuario insertado correctamente

En el caso de que al realizar alguna operación se produjese un fallo, la aplicación mostrará una pantalla advirtiéndolo del fallo como se muestra en la Figura 5.6.



Figura 5.6. Pantalla de error en una operación

Para borrar un usuario pulsamos en la opción correspondiente del menú lateral izquierdo “BORRAR USUARIO”, esta acción nos llevará a la siguiente pantalla donde nos pedirá el *login* del usuario a borrar.

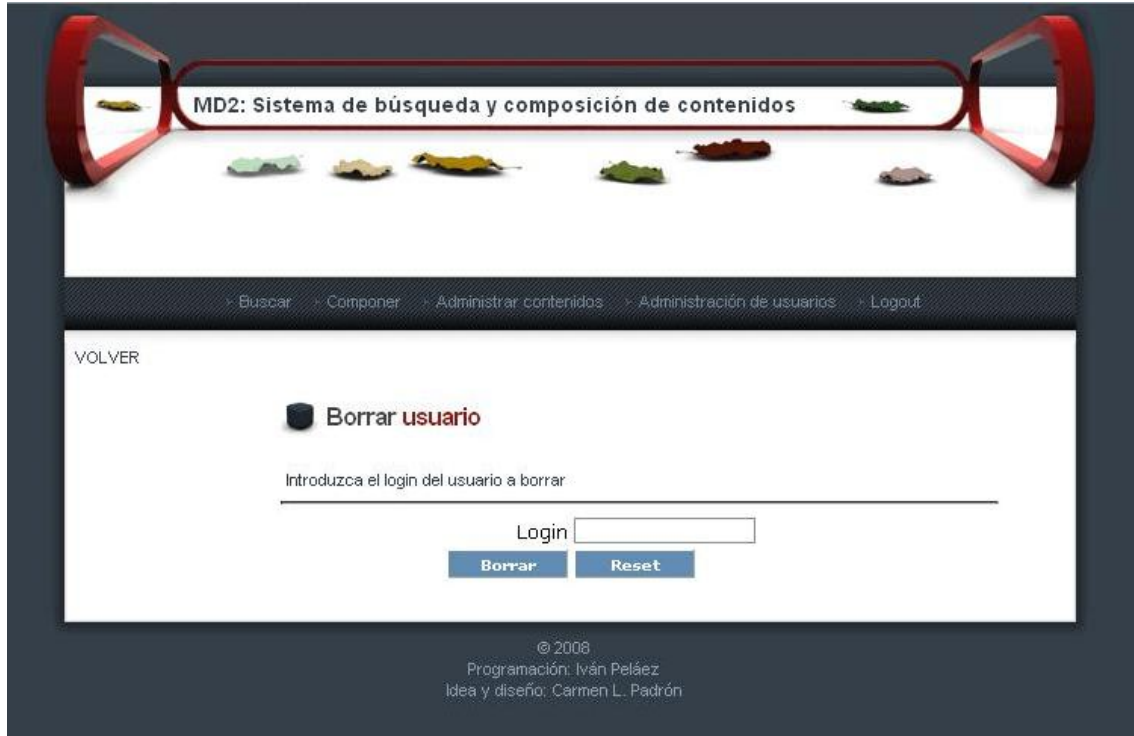


Figura 5.7. Página para buscar usuario a eliminar

Tras introducir el login del usuario y pulsar el botón “Aceptar” el sistema localizará el usuario que se intenta borrar y mostrará los datos de dicho usuario en una pantalla diferente para que el usuario administrador confirme el borrado o no.



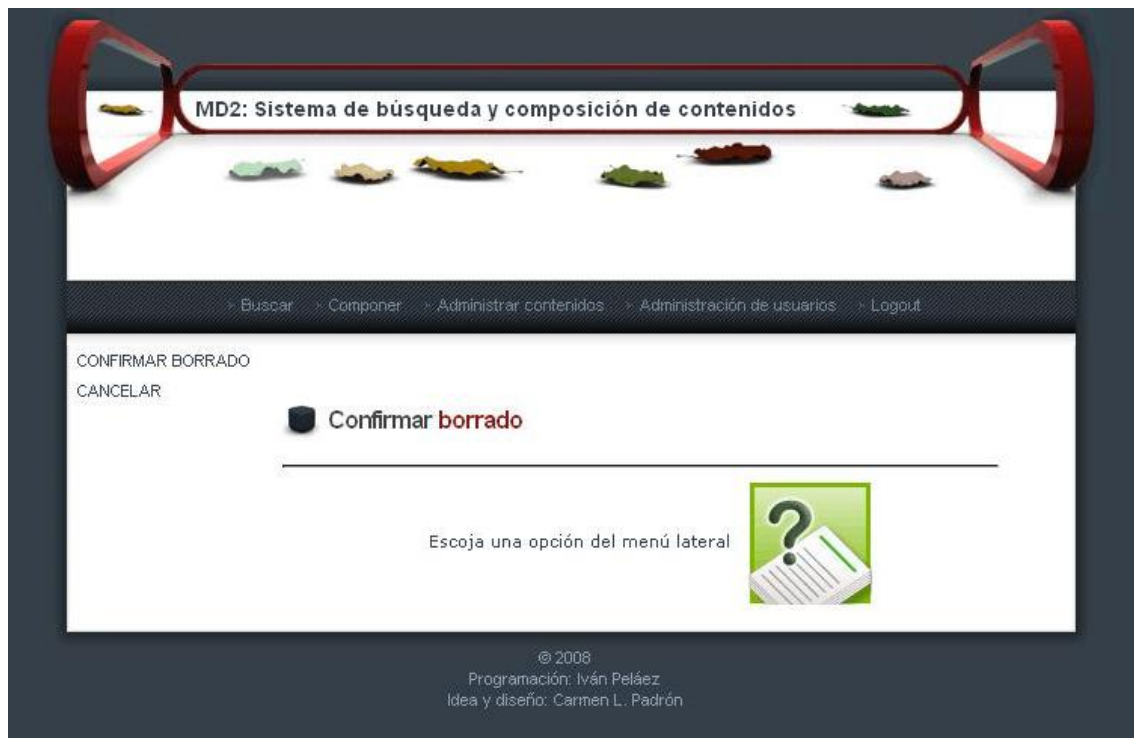


Figura 5.8. Página para confirmar el borrado de un usuario

Si el usuario confirma el borrado pulsando la opción "CONFIRMAR BORRADO" y no se produce ningún fallo el usuario se eliminará del repositorio del sistema. También se puede cancelar la operación y volver a la página de Administrar Usuarios.



Figura 5.9. Página de usuario borrado correctamente

Para modificar los datos de un usuario, se ha de pulsar en la opción “MODIFICAR USUARIO”, esta acción nos lleva a una pantalla donde se pide la introducción del login del usuario a modificar.



Figura 5.10. Página para buscar el usuario a modificar

Una vez introducido el usuario, pulsamos el botón “Aceptar” y el sistema localizará el usuario a modificar, permitiendo únicamente modificar los datos de la password y del rol, ya que el *login* ha de ser único para que pueda identificar unívocamente al usuario.

MD2: Sistema de búsqueda y composición de contenidos

> Buscar contenidos > Componer contenidos > Administrar contenidos > Administración de usuarios > Logout

[VOLVER](#)

### Modificar usuario

El usuario a modificar es el siguiente:

**Login:** usuario1  
**Rol:** Administrador

Introduzca los nuevos datos (no es posible modificar el login del usuario)

Password

Rol

[Modificar](#) [Reset](#)

© 2008  
Programación: Iván Peláez  
Ideas y diseño: Germán Rodríguez

Figura 5.11. Página para modificar los datos del usuario

Tras la introducción de los datos modificados pulsamos el botón “Modificar” almacenando los nuevos datos de usuario y dirigiéndonos a una pantalla de éxito de la operación.

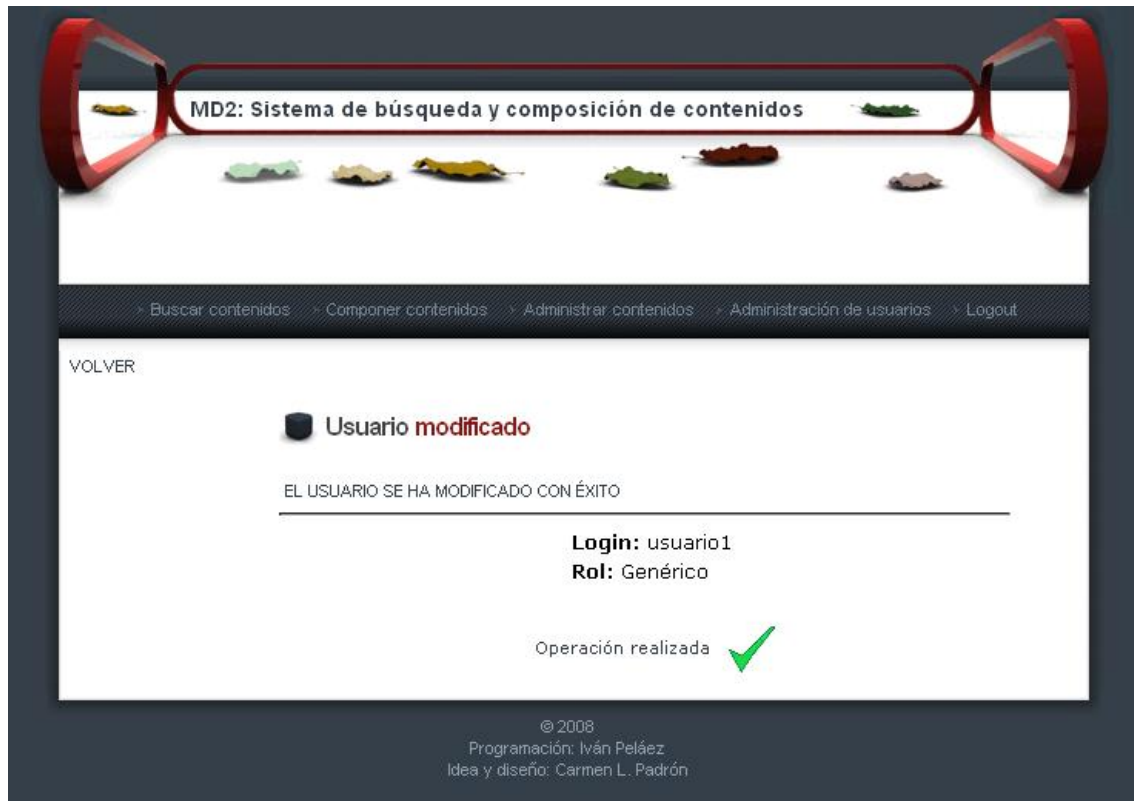


Figura 5.12. Página de usuario modificado correctamente

Por último, es posible listar todos los usuarios almacenados en el sistema pulsando sobre la opción "LISTA DE USUARIOS", esta acción nos dirigirá a otra pantalla donde se mostrará la lista de los usuarios almacenados.

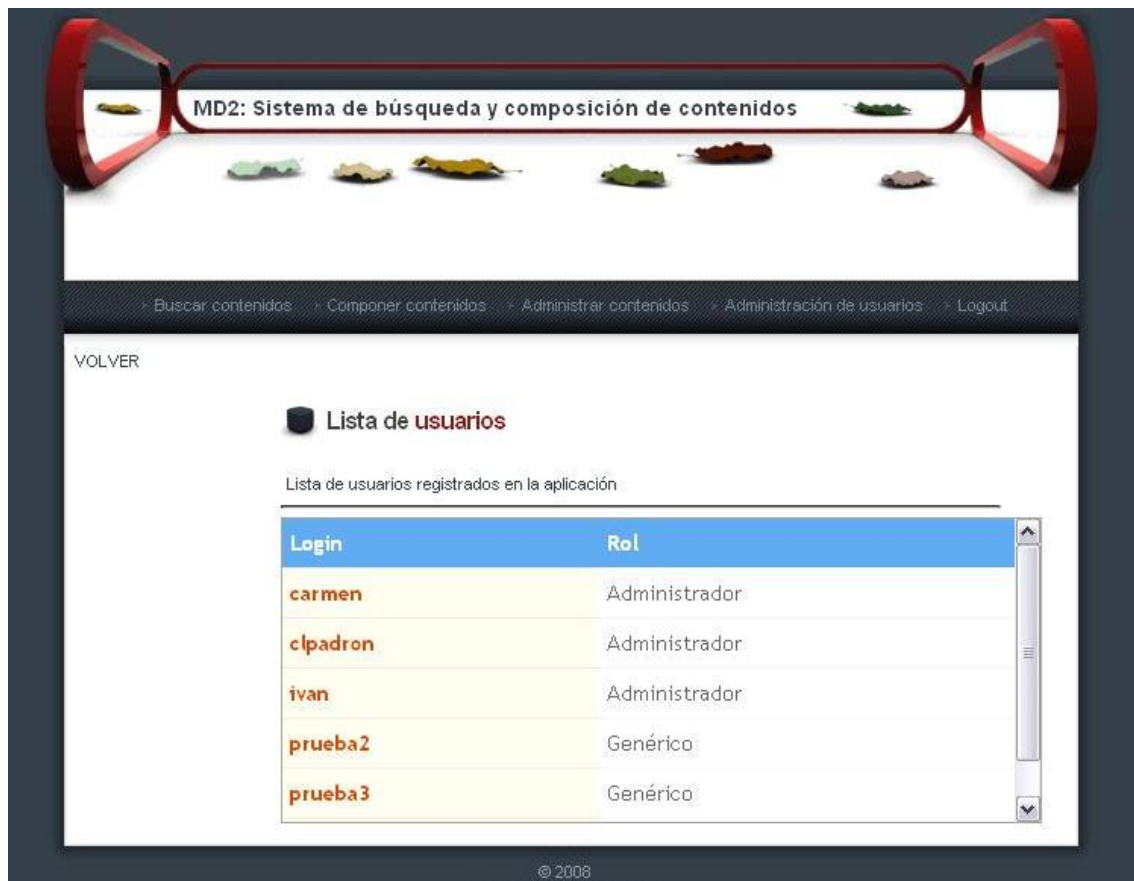


Figura 5.13. Página que muestra el listado de usuarios

### 6.3. Administrar contenidos

En esta parte de la aplicación es posible subir nuevos contenidos educativos y eliminarlos. La funcionalidad de eliminación de contenidos está limitada para el usuario con rol Genérico, ya que sólo podrá eliminar contenidos que haya subido él, y no el resto de contenidos de los demás usuarios. En cambio para el usuario con rol Administrador esta limitación no existe pudiendo eliminar cualquier contenido de cualquier usuario.



Figura 5.14. Página para administrar los contenidos

Para eliminar un contenido pulsamos la opción del menú lateral izquierdo "BORRAR CONTENIDO", esta acción nos llevará a una pantalla donde deberemos introducir el nombre del tema del contenido a borrar.



Figura 5.15. Buscador de contenidos a borrar

Una vez introducido el dato que nos piden, si pulsamos el botón “Buscar” el sistema localizará todos los contenidos que coincidan con el nombre del tema a buscar y mostrará un listado con todos los contenidos localizados. Si el usuario opta por no introducir ningún nombre de tema, el sistema mostrará el listado con todos los contenidos almacenados en el sistema, teniendo en cuenta el rol del usuario.



Figura 5.16. Listado de contenidos que se pueden borrar

Del listado mostrado el usuario podrá escoger el número de contenidos que desee para eliminarlos del sistema, para ello deberá marcar la casilla o *check box* del elemento correspondiente y pulsar el botón “Borrar”.

En este listado también se permite la opción, por cada elemento, de poder visualizar el contenido o descargarlo, que explicaremos con más detenimiento en el apartado de las búsquedas. De momento la opción Visualizar facilitará al usuario reconocer y seleccionar el contenido que desea borrar y la opción Descargar, le permitirá hacer una copia personal del contenido antes de proceder a su borrado del sistema.

Una vez que se han marcado aquellos contenidos a borrar, y se ha pulsado el botón “Borrar” el sistema nos llevará a una página de confirmación por si no



estamos seguros del borrado. Simplemente pulsando la opción “CONFIRMAR BORRADO” eliminaremos los contenidos elegidos.



Figura 5.17. Página para confirmar el borrado de los contenidos

Para subir un contenido nuevo, podemos elegir la opción “SUBIR UN CONTENIDO” en la pantalla mostrada en la Figura 5.12. Esta acción nos llevará a una página donde encontramos un pequeño formulario para introducir el nombre de tema del contenido y el fichero comprimido en formato *zip* con los archivos necesarios para la visualización de dicho contenido. El fichero ha de estar obligatoriamente en formato *zip*, para que la acción sea viable, y ha de contener un fichero *xml* (*qti.xml*) que será el examen, según la especificación IMS QTI. Si alguna de estas restricciones no se cumple, se produce un error y el sistema avisará de que no se ha podido completar la subida.



The screenshot shows a web application interface for 'MD2: Sistema de búsqueda y composición de contenidos'. At the top, there is a navigation bar with links: 'Buscar', 'Componer', 'Administrar contenidos', 'Administración de usuarios', and 'Logout'. Below this, a 'VOLVER' link is visible. The main section is titled 'Subir un contenido' with a small icon of a document. Below the title, a prompt says 'Introduzca el nombre de tema y suba el fichero'. There are two input fields: 'Nombre de Tema' and 'Fichero de contenidos'. To the right of the 'Fichero de contenidos' field is a button labeled 'Examinar...'. Below these fields are two buttons: 'Añadir' and 'Limpiar'. At the bottom of the page, there is a copyright notice: '© 2008', 'Programación: Iván Peláez', and 'Idea y diseño: Carmen L. Padrón'.

Figura 5.18. Formulario par subir un contenido

Una vez introducidos los datos pedidos, pulsamos el botón “Añadir”, hecho que nos dirige a una pantalla donde nos da la opción de subir un fichero de descriptors ya creado o crear desde cero nuestros descriptors para el contenido subido. También es posible cancelar la operación.



Figura 5.19. Menú para añadir un descriptor al contenido

Si elegimos la opción de subir un descriptor, nos dirigiremos a una página con un pequeño formulario donde podremos introducir el fichero a subir. Este fichero ha de ser un fichero xml, y que cumpla el estándar IMS Learning Resources Meta-data, en caso contrario el sistema avisará de que no se ha podido completar la operación.

The screenshot shows a web application interface for 'MD2: Sistema de búsqueda y composición de contenidos'. At the top, there is a navigation bar with links: > Buscar, > Componer, > Administrar contenidos, > Administración de usuarios, and > Logout. Below the navigation bar, there is a section titled 'VOLVER' with a sub-header 'Subir un descriptor'. The main area contains a file selection interface with the text 'Seleccione el fichero a subir' and a file input field labeled 'Fichero descriptor'. To the right of the input field is a button labeled 'Examinar...'. Below the input field are two buttons: 'Subir' and 'Reset'. At the bottom of the interface, there is a copyright notice: '© 2008', 'Programación: Iván Peláez', and 'Idea y diseño: Carmen L. Padrón'.

Figura 5.20. Formulario para subir un fichero de descriptores

Si por el contrario elegimos la opción de crear nuestros descriptores pulsando la opción "CREAR DESCRIPTOR" nos dirigiremos a una página con un formulario donde podremos asignarle una serie de descriptores al contenido. Estos descriptores están en formato combo de opciones para facilitar la validación de los mismos y como resultado de esta operación se le asignará un conjunto de descriptores que cumplen con el estándar IMS Learning Resources Meta-data al material que se ha subido.

MD2: Sistema de búsqueda y composición de contenidos

[Buscar](#)
[Componer](#)
[Administrar contenidos](#)
[Administración de usuarios](#)
[Logout](#)

[VOLVER](#)

### Crear mi propio descriptor

Introduzca uno o varios datos de su descriptor

Título: Seleccione una opción  
 Formato: Seleccione una opción  
 Duración: hh:mm:ss  
 Tipo de material: Seleccione una opción  
 Tipo de interacción: Seleccione una opción  
 Nivel de interacción: Seleccione una opción  
 Dificultad: Seleccione una opción  
 Tiempo estimado de realización: hh:mm:ss  
 Tipo de competencia: Seleccione una opción  
 Nombre de competencia:   
 Conceptos a tratar:   
 Nota de corte: Seleccione una opción

Figura 5.21. Formulario para crear descriptores del contenido

MD2: Sistema de búsqueda y composición de contenidos

[Buscar](#)
[Componer](#)
[Administrar contenidos](#)
[Administración de usuarios](#)
[Logout](#)

[VOLVER](#)

### Crear mi propio descriptor

Introduzca uno o varios datos de su descriptor

Título: Seleccione una opción  
 Formato: Seleccione una opción  
 Duración: hh:mm:ss  
 Tipo de material: Seleccione una opción  
 Tipo de interacción: Seleccione una opción  
 Nivel de interacción: Seleccione una opción  
 Dificultad: Seleccione una opción  
 Tiempo estimado de realización: hh:mm:ss  
 Tipo de competencia: Seleccione una opción  
 Nombre de competencia:   
 Conceptos a tratar:   
 Nota de corte: Seleccione una opción

Figura 5.22. Formulario para crear descriptores con un combo desplegado

Tanto si hemos elegido subir un fichero con los descriptores, como si hemos creado unos propios, llegaremos a una página de confirmación de toda la operación, es decir, que nos pide confirmar tanto la subida del fichero de contenidos como la subida/creación de los descriptores. En este punto también es posible cancelar la operación.

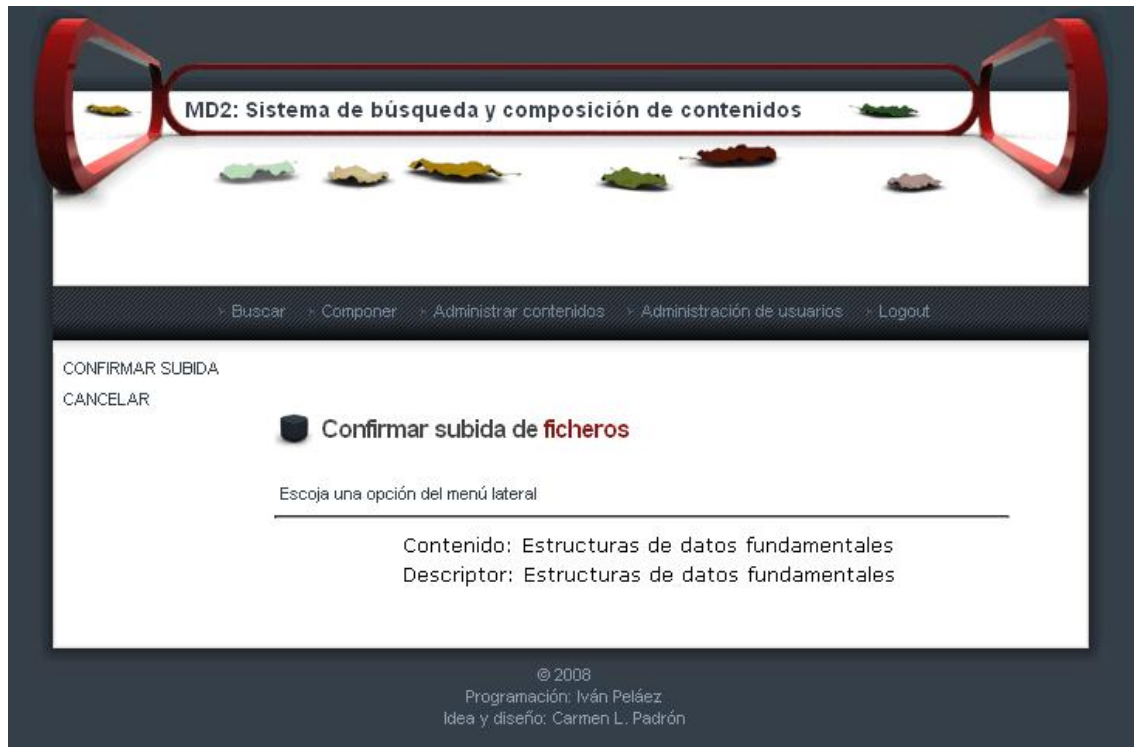


Figura 5.23. Página de confirmación para la subida de ficheros

Si el proceso ha sido satisfactorio, el sistema nos avisará que la operación se ha realizado correctamente.

### 6.3. Búsqueda de contenidos

Eligiendo esta opción en el menú principal vamos a una pantalla donde nos encontramos un buscador. En este buscador hay distintos parámetros de búsqueda que se corresponden con los posibles descriptores de un contenido. Estos descriptores vienen en formato de combos para facilitar la selección y validación de los datos.

Buscador de contenidos

Introduzca uno o varios datos del contenido

VOLVER

Titulo: Estructuras de datos fundamentales

Formato: Seleccione una opción

Tipo de material: Seleccione una opción

Tipo de interacción: Seleccione una opción

Nivel de interacción: Seleccione una opción

Dificultad: Seleccione una opción

Duración: hh mm ss

Tipo de competencia: Evaluación

Nombre de competencia: Seleccione una opción

Conceptos a tratar: Seleccione una opción

Nota de corte: Seleccione una opción

Buscar Limpiar

Figura 5.24. Buscador de contenidos educativos

Una vez que el usuario ha elegido los descriptores que considera oportunos, al presionar el botón **Buscar** nos dirigiremos a una siguiente pantalla donde se muestra un listado con los contenidos que se corresponden a los parámetros de búsqueda indicados. En el caso de la Figura 5.23 se han elegido dos parámetros de búsqueda: la temática, en este caso “Estructuras de datos fundamentales” y el tipo de competencia, en este caso “Evaluación”. Y en la Figura 5.24 se muestra el resultado que se obtendría de esta búsqueda.



Figura 5.25. Listado de contenidos buscados

En esta página se pueden realizar varias operaciones:

- Visualizar un contenido. Haciendo *click* sobre el icono de visualizar se nos abrirá una ventana distinta que mostrará el contenido en formato de página HTML. Se puede ver un ejemplo en la Figura 5.24.

CERRAR VENTANA

AVL TEST

PREGUNTAS

PREGUNTA 1

Rellene el hueco en blanco con la respuesta que estime oportuna.

El  en un árbol AVL nunca debería llegar a valer un valor absoluto superior a 1.

Mostrar respuesta

PREGUNTA 2

Rellene el hueco en blanco con la respuesta que estime oportuna.

Los árboles AVL mantienen un equilibrio en , al contrario de los árboles perfectamente equilibrados que mantienen un equilibrio mucho más estricto e ineficiente.

Mostrar respuesta

PREGUNTA 3

Califique de verdadera o falsa la siguiente afirmación.

Los árboles de Fibonacci son un caso concreto de árbol AVL.

Figura 5.26. Ejemplo de un contenido visualizado

- Descargar un contenido. Haciendo *click* sobre el icono de descargar, podremos descargarnos el contenido en formato zip. Dentro de este zip estarán todos los archivos necesarios para la visualización de ese contenido.



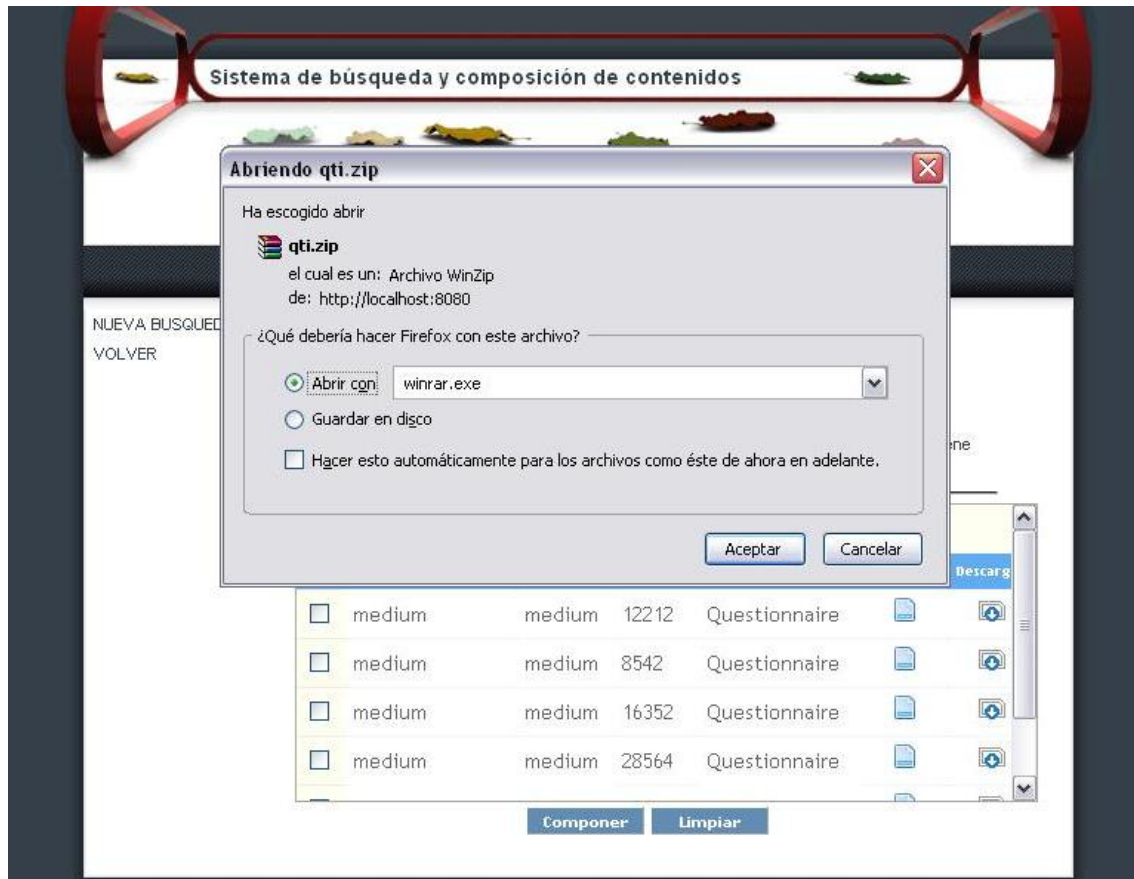


Figura 5.27. Ejemplo de descarga de contenidos

- Componer un contenido. Se puede componer contenidos en base a otros ya existentes. Para ello, el usuario deberá marcar la casilla o *check box* de los contenidos que considere oportunos que deben formar parte de la composición y a continuación deberá pulsar el botón "Componer". El sistema compondrá un nuevo contenido en base a los seleccionados y le asignará por defecto los descriptores correspondientes a los parámetros de búsqueda, para que este contenido pueda ser localizado posteriormente. Esta acción nos llevará a una nueva pantalla donde se puede ver el nuevo contenido compuesto y donde a su vez podemos visualizar o descargar dicho contenido.



Figura 5.28. Página donde se muestra el nuevo contenido compuesto

- Nueva búsqueda. También podrá realizar una nueva búsqueda simplemente pulsando en la opción del menú lateral izquierdo "NUEVA BÚSQUEDA".

#### 5.4. Componer contenidos

Esta opción es muy similar a la Búsqueda de contenidos, ya que dicha opción nos dirigirá a un buscador similar al de la Figura 5.24 donde podremos localizar los materiales para componer un contenido nuevo.

Una vez localizados, el proceso es similar al anteriormente explicado, se podrán marcar varios contenidos del listado mostrado (al menos dos), y pulsando sobre el botón "Componer" compondremos el nuevo contenido que quedará almacenado junto con los parámetros de búsqueda utilizados, ya que éstos serán usados como descriptores del contenido. Como en el caso de las búsquedas, también es posible la descarga y visualización de todos los contenidos mostrados en el listado.

Una vez que el contenido ha sido compuesto, será mostrado en una página distinta y podrá ser visualizado o descargado por el usuario.

### 5.5. Logout del sistema

Por último, pinchando en la opción del menú principal "Logout" finalizaremos la sesión con el sistema volviendo a la página inicial de la aplicación, que no es otra, que el formulario de identificación (Figura 6.1.).

### 5.6. Configuración de la aplicación

Para terminar este capítulo, se incluye unas instrucciones básicas para configurar la aplicación en un contenedor web. En este caso Apache Tomcat 5.5.

Partimos de la base de que el servidor ya está instalado, junto con el gestor de base de datos, en este caso MySQL 5.0.

Para que el servidor pueda arrancar la aplicación hay que almacenar el módulo WAR (Web-Archives) de la aplicación en la carpeta *webapps* del servidor y configurar el *DataSource* necesario para conexión con la base de datos. Este *DataSource* se especifica en el fichero *server.xml* del Tomcat, que se encuentra en la carpeta *conf* del servidor.

Para configurar el *DataSource* incluimos en el fichero citado las siguientes líneas de código XML, dentro de la etiqueta *<host>*:

```
<Context path="/qticomposer" docBase="[Ruta_base_del_servidor]\apache-  
tomcat-5.5.20\webapps\qticomposer" debug="0" reloadable="true">  
  <Resource name="jdbc/bdproyecto" auth="Container"  
    type="javax.sql.DataSource"  
    maxActive="100" maxIdle="30" maxWait="10000"  
    username="usuario_bbdd" password="password_usuario"  
    driverClassName="com.mysql.jdbc.Driver"  
    url="jdbc:mysql://localhost:3306/[nombre_base_de_datos]"/>  
</Context>
```

Vemos que en el atributo *path* se incluye la carpeta raíz del proyecto, y en el atributo *docBase* la ruta de instalación de la aplicación, es decir, la ruta de instalación del servidor Tomcat, más la ruta dónde se encuentra la aplicación.

Dentro de la etiqueta *Resource*, se configuran los datos de conexión con la base de datos, incluyendo la url de la base de datos y el usuario y la password que hayamos creado en la base de datos para poder acceder a ella.

Para terminar se incluye el script de creación de la base de datos.

```
CREATE DATABASE [nombre_base_de_datos];
USE [nombre_base_de_datos];
DROP USER [usuario_bbdd];
CREATE USER [usuario_bbdd] IDENTIFIED BY '[usuario_bbdd]';
GRANT ALL PRIVILEGES ON *.* TO [usuario_bbdd];

DROP TABLE relacion_cursos;
DROP TABLE cursos;
DROP TABLE ITEMS;
DROP TABLE DESCRIPTORES;
DROP TABLE USUARIOS;

CREATE TABLE DESCRIPTORES (
  id_descriptor int(11) NOT NULL auto_increment,
  usuario varchar(20) default NULL,
  fichero blob,
  title varchar(100) default NULL,
  language varchar(15) default NULL,
  format varchar(15) default NULL,
  duration int(11) default NULL,
  learningResourcetype varchar(30) default NULL,
  interactivitytype varchar(30) default NULL,
  interactivitylevel varchar(30) default NULL,
  difficulty varchar(30) default NULL,
  typicallearningtype int(11) default NULL,
  incumbentname varchar(100) default NULL,
  incumbenttype varchar(100) default NULL,
  concepts varchar(200) default NULL,
  mark int(11) default NULL,
  CONSTRAINT PK_DESCRIPTORES PRIMARY KEY (id_descriptor),
  CONSTRAINT FK_DESCRIPTORES FOREIGN KEY (usuario) REFERENCES
  USUARIOS(login)
  ON DELETE SET NULL
  ON UPDATE CASCADE
);

CREATE TABLE ITEMS (
  id_item int(11) NOT NULL auto_increment,
  nombre varchar(50) NOT NULL,
  fichero blob NOT NULL,
```

```

usuario varchar(20) default NULL,
CONSTRAINT PK_ITEMS PRIMARY KEY (id_item),
CONSTRAINT FK_ITEMS FOREIGN KEY (usuario) REFERENCES USUARIOS(login)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE CURSOS (
    curso int(11) NOT NULL,
    item int(11) NOT NULL,
    CONSTRAINT PK_CURSOS PRIMARY KEY (id_curso),
    CONSTRAINT FK_CURSOS FOREIGN KEY (usuario) REFERENCES
USUARIOS(login)
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE RELACION_CURSOS(
    curso INTEGER NOT NULL,
    item INTEGER NOT NULL,
    CONSTRAINT PK_RELACION_CURSOS PRIMARY KEY (curso,item),
    CONSTRAINT FK1_RELACION_CURSOS FOREIGN KEY (curso) REFERENCES
CURSOS(id_curso)      ***** creada
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT FK2_RELACION_CURSOS FOREIGN KEY (item) REFERENCES
ITEMS(id_item)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE USUARIOS(
    login varchar(20) NOT NULL,
    password varchar(15) NOT NULL,
    rol varchar(1) NOT NULL,
    CONSTRAINT PK_USUARIOS PRIMARY KEY(login)
);

CREATE TABLE DESCRIBEN(
    descriptor INTEGER NOT NULL,
    item INTEGER NOT NULL,
    CONSTRAINT PK_DESCRIBEN PRIMARY KEY (descriptor,item),
    CONSTRAINT FK1_DESCRIBEN FOREIGN KEY (descriptor) REFERENCES
DESCRIPTORES(id_descriptor)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT FK2_DESCRIBEN FOREIGN KEY (item) REFERENCES
ITEMS(id_item)
    ON DELETE CASCADE

```

```
        ON UPDATE CASCADE
    );

CREATE TABLE BUSCA(
    id_búsqueda INTEGER NOT NULL auto_increment,
    usuario VARCHAR(20) DEFAULT 'nouser',
    item INTEGER NOT NULL,
    marcado INTEGER CHECK marcado IN (0|1),
    CONSTRAINT PK_BUSCA PRIMARY KEY (id_búsqueda),
    CONSTRAINT FK1_BUSCA FOREIGN KEY (usuario) REFERENCES
USUARIOS(login)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT FK2_BUSCA FOREIGN KEY (item) REFERENCES ITEMS(id_item)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

---

## 6. GESTIÓN DEL PROYECTO

---

La planificación del proyecto se ha realizado teniendo en cuenta una metodología en cascada y por tanto la realización de las tareas, en las que ha sido dividido el proyecto, se efectúan de manera sucesiva, es decir, cuando acaba una, comienza la siguiente. En la Figura 4.1 se muestra con detalle la división de tareas planteada para el proyecto, la duración de cada tarea, así como la fecha de comienzo y final de cada tarea.

El proyecto se inició en Febrero de 2007 y concluyó en Julio de 2008, encontrándose mayores dificultades en las tareas de implementación de los módulos de visualización de contenidos y de composición de contenidos.

En la planificación del proyecto no se ha incluido ninguna gestión de riesgos, dado el carácter académico del mismo.

	i	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Recogida de Requisitos	20 días	lun 02/07/07	vie 27/07/07	
2		<b>- Análisis</b>	<b>30 días</b>	<b>lun 30/07/07</b>	<b>vie 07/09/07</b>	<b>1</b>
3		Análisis de Requisitos	12 días	lun 30/07/07	mar 14/08/07	
4		Definición de los casos de uso	7 días	mié 15/08/07	jue 23/08/07	3
5		Casos de uso extendidos	5 días	vie 24/08/07	jue 30/08/07	4
6		Diagramas de actividad	6 días	vie 31/08/07	vie 07/09/07	5
7		Fin de Análisis	0 días	vie 07/09/07	vie 07/09/07	2
8		<b>- Diseño</b>	<b>28 días</b>	<b>lun 10/09/07</b>	<b>mié 17/10/07</b>	<b>7</b>
9		Modelo de Datos	8 días	lun 10/09/07	mié 19/09/07	
10		Modelo de clases	10 días	jue 20/09/07	mié 03/10/07	9
11		Diagrama de secuencia	4 días	jue 04/10/07	mar 09/10/07	10
12		Diagramas de colaboración	6 días	mié 10/10/07	mié 17/10/07	11
13		Fin Diseño	0 días	mié 17/10/07	mié 17/10/07	8
14		<b>- Implementación</b>	<b>136 días</b>	<b>jue 18/10/07</b>	<b>jue 24/04/08</b>	<b>13</b>
15		Script Base de Datos	15 días	jue 18/10/07	mié 07/11/07	
16		Módulo Administración de Usuarios	18 días	jue 08/11/07	lun 03/12/07	15
17		Módulo Gestión de contenidos	17 días	mar 04/12/07	mié 26/12/07	16
18		Módulo Gestión de descriptores	15 días	jue 27/12/07	mié 16/01/08	17
19		Módulo Búsqueda de contenidos	20 días	jue 17/01/08	mié 13/02/08	18
20		Módulo Visualización de contenidos	26 días	jue 14/02/08	jue 20/03/08	19
21		Módulo Composición de contenidos	25 días	vie 21/03/08	jue 24/04/08	20
22		Fin Implementación	0 días	jue 24/04/08	jue 24/04/08	14
23		Pruebas	10 días	vie 25/04/08	jue 08/05/08	22
24		Implantación	2 días	vie 09/05/08	lun 12/05/08	23
25		Documentación del Proyecto	134 días	mar 13/05/08	vie 14/11/08	24
26						

Figura 6.1. Planificación de tareas del proyecto

Las tareas llevadas a cabo son las siguientes:

- Recogida de requisitos. En esta tarea previa se han mantenido reuniones con la tutora del proyecto, que en este caso actuaba de cliente, para definir la funcionalidad de la aplicación de manera que quedaran claros los requisitos que debería cumplir la aplicación.
- Análisis. En esta tarea se han realizado varias actividades, todas ellas comprendidas en la fase de análisis. Estas actividades son las necesarias para analizar el problema a partir de los requisitos de la fase previa. Como resultado de este análisis se han definido los diagramas de casos de uso, junto con los casos de uso extendidos y los diagramas de actividad que nos ayudarán a entender el comportamiento del sistema.
- Diseño. Dentro de esta actividad se han realizado varias actividades que han dado como resultado la creación de varios diagramas. La primera actividad es la creación del modelo de datos que ha dado como resultado el diagrama Entidad/Relación y el modelo relacional. A continuación se ha definido el diagrama de clases junto con los diagramas de secuencia y de colaboración, que ha permitido tener una



visión de cómo se iban a hacer las cosas, y del comportamiento dinámico de la aplicación.

- Implementación. En esta tarea se ha llevado a cabo toda la codificación de la aplicación. En primer lugar se realizó el script de base de datos para la creación física de la base de datos y a continuación se implementaron cada uno de los módulos en los que está separada la aplicación.
- Pruebas. En esta fase se probaron cada uno de los módulos para comprobar el correcto funcionamiento de todas las partes de la aplicación. Fuera de esta fase la aplicación ha sido utilizada, y por tanto probada, por parte de los alumnos de la asignatura “Enseñanza asistida por ordenador” del quinto curso de la titulación Ingeniería Informática.
- Implantación. En esta tarea se han llevado a cabo las acciones necesarias para poner en funcionamiento en un entorno de producción la aplicación. Para ello se ha configurado el contenedor de aplicaciones y el gestor de base de datos que iban a acoger tanto la aplicación como la base de datos, respectivamente.
- Documentación del proyecto. Por último se ha plasmado en este documento todo el trabajo realizado durante todas las fases del proyecto y el estudio realizado sobre el estado actual del ámbito e-learning.

En la Figura 6.2. se muestra el diagrama Gantt relativo al proyecto, en él se pueden ver las diferentes tareas a lo largo de un tiempo total determinado. También se muestra el tiempo previsto para cada tarea y las relaciones existentes entre las tareas.

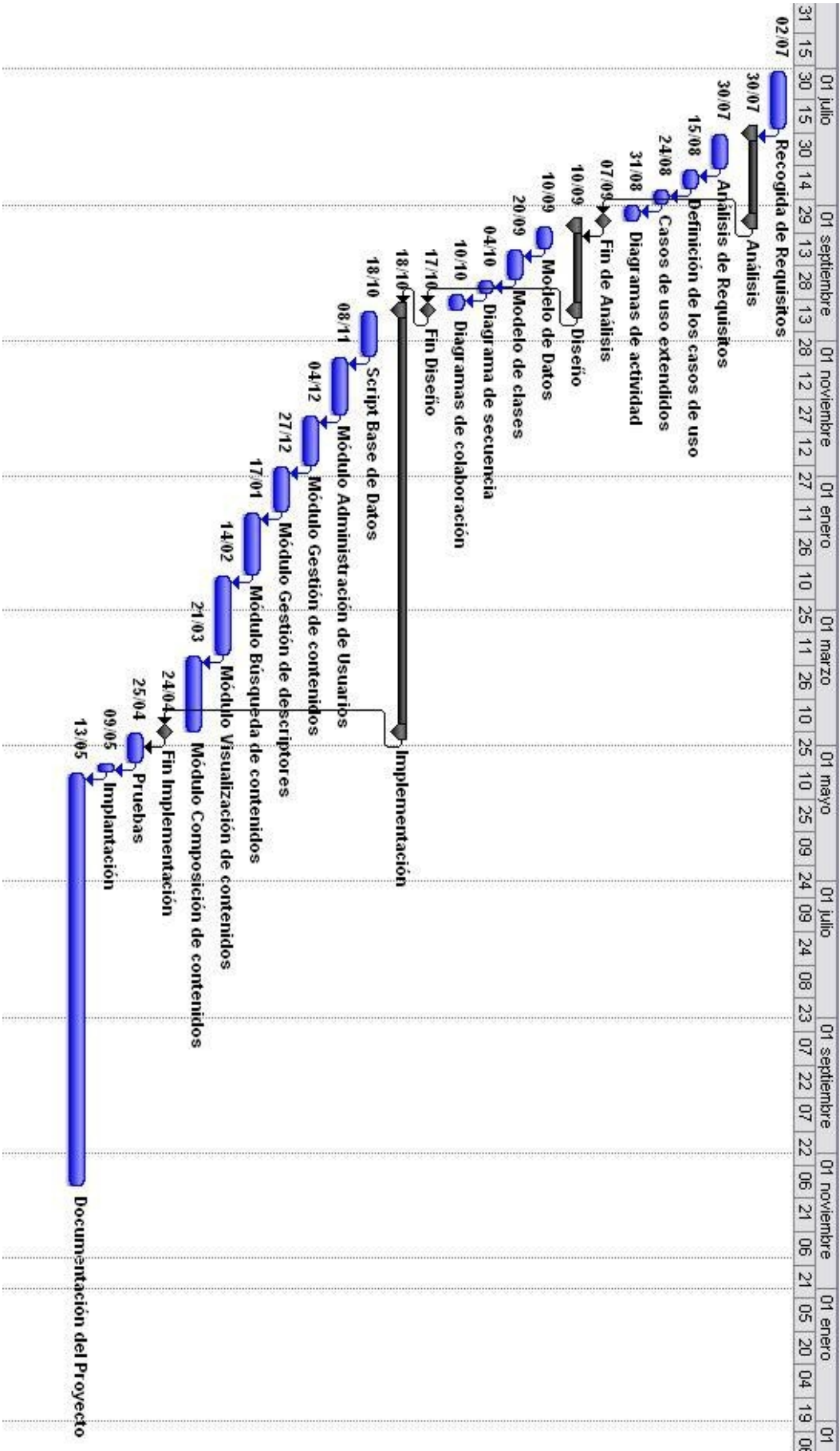


Figura 6.2. Diagrama Gantt del proyecto

## 6.1 *Análisis de costes*

Para analizar los costes que ha supuesto el desarrollo de la aplicación hemos dividido estos en grupos, quedando éstos organizados de la siguiente manera: personal implicado, material e infraestructuras utilizadas y otros.

### 6.1.1 Personal

Para el desarrollo de la aplicación han sido necesarios varios perfiles de personal, entre ellos un analista funcional, un analista orgánico y un analista/programador. Se ha prescindido del perfil de Jefe de proyecto ya que realmente el equipo de trabajo solo constaba de una persona debido al carácter académico del proyecto y sus tareas se han agrupado dentro del mismo perfil al analista funcional y al analista orgánico. Por tanto para analizar los costes de personal solo se tienen en cuenta dos perfiles: el analista funcional (que también realizará las tareas de analista orgánico) y el analista/programador.

A continuación detallamos los días que cada perfil ha trabajado en cada actividad para el desarrollo final de la herramienta:

Tarea / N° días	Analista funcional/orgánico	Analista- Programador
Recogida de Requisitos	20	0
<b>Análisis</b>		
Análisis de Requisitos	12	0
Definición de los casos de uso	7	0
Casos de uso extendidos	5	0
Diagramas de actividad	6	0
<b>Diseño</b>		
Modelo de Datos	8	2
Modelo de clases	10	8
Diagrama de secuencia	4	4

Diagramas de colaboración	6	6
<b>Implementación</b>		
Script Base de Datos	3	15
Módulo Administración de usuarios	0	18
Módulo Gestión de contenidos	0	17
Módulo Gestión de descriptores	0	15
Módulo Búsqueda de contenidos	0	20
Módulo Visualización de contenidos	0	26
Módulo Composición de contenidos	0	25
<b>Pruebas</b>	10	10
Implantación	0	2
<b>TOTAL:</b>	<b>91</b>	<b>168</b>

Figura 6.3. Costes del personal implicado en el proyecto

El resumen de días y costes totales se muestra en la siguiente tabla:

	<b>Días trabajados</b>	<b>€/ Día</b>	<b>Coste Total</b>
<b>Analista Funcional/Orgánico</b>	91	120	10920 €
<b>Analista - Programador</b>	168	80	13440 €
		<b>TOTAL:</b>	<b>24360 €</b>

Figura 6.4. Resumen de días y costes totales del personal

### 6.1.2 Material e infraestructuras

Para el desarrollo de la aplicación también ha sido necesario el uso de material, ya sea desde el hardware utilizado para el desarrollo de la herramienta como la cantidad de papel que ha hecho falta para todas las

notas tomadas, o la conexión a Internet. Por ello he realizado la siguiente tabla indicando los costes de cada uno:

	Precio	Cantidad	Subtotal
<b>Portátil HP compaq nx5000</b>	1199	1	1199
<b>Licencia de Microsoft Windows XP</b>	132	1	132
<b>CD-R 80 min</b>	0,50	1	0,50
<b>Papel A5 (500 hojas)</b>	3	1	3
<b>Tinta impresora</b>	30	1	30
<b>Conexión Internet</b>	30	4	120
		<b>IVA 16%</b>	<b>237,52 €</b>
		<b>TOTAL:</b>	<b>1722,02 €</b>

Figura 6.5. Costes de materiales e infraestructuras

### 6.1.3 Otros

Por último, tendríamos los costes que no se podrían englobar en ninguno de los otros grupos. Siguiendo el mismo método que con los anteriores, he realizado una tabla para indicar estos costes:

	Precio	Cantidad	Subtotal
<b>Abono Transporte</b>	47	3	141
<b>Gasolina diesel (40 litros)</b>	40	2	80
		<b>TOTAL:</b>	<b>221 €</b>

Figura 6.6. Otros gastos

#### 6.1.4 Resumen de costes

Para terminar, voy a hacer un balance del coste total de la aplicación. Para ello voy a sumar todos los costes que he calculado anteriormente.

	Costes
<b>Personal</b>	24360
<b>Material e Infraestructuras</b>	1722,02
<b>Otros</b>	221
<b>TOTAL:</b>	<b>26303,02 €</b>

Figura 6.7. Resumen de costes del proyecto

A continuación, y para calcular el precio final que le cobraríamos al cliente por la aplicación, voy a aplicarle el 20% de beneficios al total del coste.

$$26303,02 + 5260,60 = 31563,62 \text{ €}$$

---

## 7. CONCLUSIONES

---

Como parte final del trabajo desempeñado a lo largo de este proyecto de fin de carrera, se pueden extraer algunas conclusiones y también citar algunas ideas sobre trabajos futuros o ampliaciones al mismo, que exponemos a continuación.

### *7.1 Conclusiones del proyecto*

El presente proyecto se ha centrado en el campo del *e-learning*, más concretamente, en el movimiento estandarizador que actualmente intenta establecer un conjunto de normas que atañen al aprendizaje on-line. Por ello, algunas de las conclusiones de este proyecto se derivan del estudio previo realizado sobre la situación actual en la que se encuentra el *e-learning*. Éstas son:

- La estandarización es fundamental para conseguir un avance en el mercado del *e-learning*. Capacidades tales como la interoperabilidad y reusabilidad de cualquier objeto de aprendizaje son de alta importancia para el desarrollo del mercado de *e-learning*, tanto para software específico como para contenidos. Alcanzar un alto grado en

estas y otras capacidades es un objetivo para cuyo alcance es necesario un proceso de estandarización. En la actualidad existen un conjunto de organismos que crean especificaciones y estándares para *e-learning*. IMS Global es el organismo más activo y prolífico en esta materia, aunque existen otros organismos dedicados también a esta tarea. De ahí que este proyecto se haya centrado en estándares de dicho organismo.

- Existe una problemática a la hora de desarrollar contenidos didácticos y es la falta de medios para describir los componentes y lograr crear contenidos reutilizables, por ello en este proyecto hemos abordado este problema aplicando una posible solución, que es la utilización del método creado para tal efecto, llamada MD2 (Método de Desarrollo de Materiales Didácticos). Con este método, se describen de forma general los requisitos del contenido, pedagógicos, de soporte técnico y de calidad de los materiales didácticos a través de descriptores y se define como proceder durante el desarrollo.
- Con una descripción de los contenidos completa, se facilita bastante la operación de búsqueda de dichos contenidos y por tanto una mejor localización de éstos, para que el acceso sea mayor y más rápido.
- La utilización de descriptores en la definición de contenidos didácticos permite componer contenidos más personalizados, y por tanto mejorar la oferta de contenidos, mostrando al usuario la oportunidad de crear contenidos más *a su medida*.

Además de las conclusiones sobre la situación actual del *e-learning*, gran parte de este proyecto se ha dedicado también a la implementación de una herramienta, cuyo desarrollo ha permitido un mejor conocimiento de las tecnologías relacionadas con Java y que han derivado también en algunas conclusiones:

- El patrón Modelo Vista Controlador ayuda a los desarrolladores de aplicaciones empresariales a estructurar mejor el trabajo y a hacer más fácil el mantenimiento de las aplicaciones creadas con este patrón. Dividir el desarrollo en tres capas proporciona varias ventajas y no muchos inconvenientes, especialmente para grandes aplicaciones en las que es posible definir varias vistas o interfaces de usuario con una misma lógica de negocio.
- El Framework Struts es una excelente ayuda para la implementación del patrón MVC en aplicaciones Web implementadas en lenguaje Java. Este API desarrollado por la Apache Software Foundation permite de forma rápida y fácil, desarrollar las tres capas del patrón. Para implementar el modelo basta con crear algunas clases java que extienden de la clase Action, obteniendo los datos introducidos por el usuario encapsulados en objetos que extienden de la clase ActionForm, facilitando también un método de validación. La vista se implementa en JSP, para ello puede ser útil la utilización de las Tag Libraries,



ahorrando código en las páginas. Y por el controlador es suficiente con configurar de forma correcta el fichero `struts-config.xml`. Este fichero actúa como mapa, indicando qué Action ejecutar para cada petición y qué vista mostrar como resultado a su ejecución.

- El API JDOM provee un conjunto de clases que permiten una integración de Java y XML de forma fácil e intuitiva.
- Existe una gran variedad de APIs y herramientas Open Source que proporcionan numerosas y potentes funcionalidades muy interesantes, desde el software necesario para desarrollar en Java, hasta el contenedor web (Apache Tomcat) necesario para poner en producción la herramienta, pasando por APIs y entornos de desarrollo (NetBeans, Eclipse, etc.). Es posible desarrollar una aplicación utilizando estas tecnologías y herramientas sin invertir en ningún software propietario.

## *7.2 Trabajo futuro y ampliaciones*

Este proyecto fin de carrera deja abiertas algunas líneas para trabajo futuro y admite varias ampliaciones. Podemos citar algunas líneas de trabajo.

1. Extender la herramienta para admitir otras especificaciones que sí están implementadas en otras herramientas de autor, como por ejemplo ADL SCORM, e incluso admitir especificaciones más recientes de las ya implementadas de la IMS Global como IMS LD.
2. Completar la edición de los contenidos que facilite la creación desde cero de contenidos educativos así como su posible modificación, de una manera sencilla y atractiva para el usuario.
3. Desarrollar la parte de gestión de resultados de los test almacenados en la aplicación para que el usuario pudiera realizar los exámenes y obtener una puntuación acorde a sus conocimientos.
4. En relación a las búsquedas de los contenidos sería deseable ampliar el abanico a otros repositorios accesibles vía servicios web, tal como se indica en la solución MD2, de esta manera la búsqueda de contenidos no se limitaría a un repositorio local y la localización de contenidos sería más enriquecedora.
5. En cuanto a la descripción de los contenidos, se podría ampliar la restricción inicial de centrarnos en una sola área de conocimiento "Ciencias de la Computación", descrita según los planes de estudio de la ACM (Association for Computing Machinery). Extender a otras áreas de conocimiento descritas según el sistema de catalogación de la Librería del Congreso de los Estados Unidos (LLC).

6. En relación a la accesibilidad de la aplicación se podrían aprovechar las características de internacionalización del Framework Struts, para que la herramienta no se centre en un solo idioma. Además de facilitar diferentes hojas de estilo CSS que permitan la visualización de la información de la aplicación para usuarios con problemas de visión.

---

## REFERENCIAS

---

- [1] Eckel, Bruce. **Piensa en Java**. Prentice Hall, 2002
- [2] Fernández Manjón, Baltasar. **Especificaciones y estándares en e-learning**. CNICE (Red Digital 2006).
- [3] Fernández Manjón, Baltasar; Martínez Ortiz, Iván; Moreno Ger, Pablo; Sierra Rodríguez, José Luis. **Uso de estándares aplicados a Tic en educación**.  
<http://ares.cnice.mec.es/informes/16/index.htm>
- [4] Fernández Manjón, Baltasar; Sancho, Pilar. **<e-aula>: Entorno de Aprendizaje Personalizado Basado en Estándares Educativos**.
- [5] Foix, Cristian; Zavando, Sonia. **Estándares e-learning: Estado del Arte**. Corporación de Investigación Tecnológica de Chile (INTEC), 2002.  
<http://empresas.sence.cl/documentos/elearning/INTEC%20-%20Estandares%20e-learning.pdf>
- [6] Padrón C. L. (2006) **MD2 method: the Didactic Material Creation from a Model Perspective**. In Proceedings of PhD Consortium of the EC-TEL06 Conference, Crete, Greece, October 2006.  
<http://www.prolearn-academy.org/Academy%20Events/ectel/cd-proceedings>
- [7] Rosenberg, Marc Jeffrey. **E-learning: estrategias para transmitir conocimiento en la era digital**. Mc Graw Hill, 2002
- [8] Sun Microsystems, Inc. **E-learning Interoperability Standards**.  
[http://www.sun.com/solutions/documents/white-papers/ed\\_interop\\_standards.pdf](http://www.sun.com/solutions/documents/white-papers/ed_interop_standards.pdf)

### Referencias web

- [9] **ADL**, Advanced Distributed Learning.  
<http://www.adlnet.gov/>

- [10] AICC, Aviation Industry Computed Based-Training Comitee.  
<http://www.aicc.org/>
- [11] Apache Software Foundation.  
<http://www.apache.org>
- [12] ARIADNE, Alliance of Remote Instructional Authoring and Distribution Networks for Europe.  
<http://www.aicc.org/>
- [13] COMPRUEBA.  
<http://alamo.sim.ucm.es/comprueba/intro.htm>
- [14] Crimson, XML Parser.  
<http://xml.apache.org/crimson/>
- [15] IMS Content Packaging Specification.  
[http://www.imsglobal.org/content/packaging/cpv1p1p2/imsdp\\_bindv1p1p2.html](http://www.imsglobal.org/content/packaging/cpv1p1p2/imsdp_bindv1p1p2.html)
- [16] IMS Global Learning Consortium Specification.  
<http://www.imsglobal.org/>
- [17] IMS Learning Resources Meta-data Specification.  
[http://www.imsglobal.org/metadata/imsmdv1p2/imsmd\\_bindv1p2.html](http://www.imsglobal.org/metadata/imsmdv1p2/imsmd_bindv1p2.html)
- [18] IMS QTI, IMS Question & Test Interoperability Specification.  
[http://www.imsglobal.org/question/qti1p2/imsqti\\_asv1p2.html](http://www.imsglobal.org/question/qti1p2/imsqti_asv1p2.html)
- [19] JDOM, Java Document Object Model.  
<http://www.jdom.org>
- [20] .LRN (dot-learn)  
<http://dotlrn.org>
- [21] LTSC, IEEE Learning Technologies Standards Committee.  
<http://ieeeltsc.org/>
- [22] Moodle.  
<http://moodle.org>
- [23] MVC, Patrón Modelo-Vista-Controlador.  
<http://java.sun.com/blueprints/patterns/MVC-detailed.html>
- [24] Oracle's parsers.  
<http://www.oracle.com/technology/tech/xml/index.html>
- [25] Struts.  
<http://struts.apache.org>
- [26] WebCT, Web Course Tool.

<http://www.blackboard.com/us/index.bbb>

- [27] Xerces Java Parser.  
<http://xerces.apache.org/xerces-j/>
- [28] XML4j, XML Parser for Java.  
<http://www.alphaworks.ibm.com/tech/xml4j>
- [29] XSLT, XSL Transformations.  
<http://www.w3.org/TR/xslt.html>